공학석사 학위논문

비정형 데이터를 이용한 순환 신경망 기반의 재난 문장 탐지 방법

- 화재 재난을 대상으로 -

Detecting Disaster Information Sentences from Unstructured Data Using Recurrent Neural Network

- Case Study on Fire Accident -

2018년 8월

서울대학교 대학원

건설환경공학부

임 장 혁

비정형 데이터를 이용한 순환 신경망 기반의 재난 문장 탐지 방법

- 화재 재난을 대상으로 -

Detecting Disaster Information Sentences from Unstructured Data Using Recurrent Neural Network

- Case Study on Fire Accident -

지도교수 김 용 일

이 논문을 공학석사 학위논문으로 제출함

2018년 5월

서울대학교 대학원

건설환경공학부

임 장 혁

임장혁의 공학석사 학위논문을 인준함

2018년 5월

위 원 장 이 창 원 (인)

부위원장 김 용 일

위 원 유 기 준

국문초록

빅데이터는 비정형 데이터인 텍스트로 이루어져 있어 텍스트 마이닝을 통해 정책 수립, 의사 결정에 대한 유의미한 정보를 도출할 수 있는 분석이 가능하다. 텍스트 마이닝 기법 중 순환 신경망을 사용한 최근의 연구들은 기존의 딥러닝 알고리즘인 CNN 및 다른 기계학습 알고리즘보다 성능이 향상된 것을 확인할 수 있다. 딥러닝 알고리즘은 양질의 학습 데이터의 양에 따라 학습의 효율 및 결과가 달라진다. 따라서 본 연구에서는 Word2Vec 모델을 이용한 학습 데이터를 증강하는 기법을 적용하여 재난 문장탐지 모델의 정확도가 개선되는지를 확인해보았다. 또한, 순환 신경망의 종류인 LSTM과 GRU를 이용한 텍스트 분석 결과를 비교하여 소셜미디어에서의 화재 발생의 정보를 담고 있는 문장을 탐지하는 방법의 정확도를 향상하고자 한다.

본 연구에서 제안하는 재난 문장탐지 모델은 데이터를 증강하는 과정과 모델이 학습하는 단계에서 선행연구보다 사용자의 개입을 최소화하고, 정확도를 향상했다. 또한, 탐지된 재난 문장은 추후 개체명 인식을 이용하여 정형화할 수 있어 비정형 데이터에서의 재난 위치를 비롯한 정보를 추출할 수 있다는 점에서 의의가 있다.

주요어 : 딥러닝, 순환신경망, 데이터 증강, Word2Vec, 재난 정보, 소셜미디어

학 번 : 2016-24243

목 차

표 목 차

그 림 목 차

1. 서론

1.1 연구 배경 및 목적

빅데이터 분석 기술은 정책 제언 의사 결정뿐만 아니라 가까운 미래를 예측하는 등 많은 분야에서 활발히 활용되고 있으며, 2016년에는 1.6ZB의 데이터가 생성되었고, 2025년에는 이의 10배에 달하는 데이터가 생성될 것이라고 할 정도로 규모가 급속도로 성장하고 있다(Reinsel, 2017). 특히 일본에서 2011년 발생한 도호쿠 지진 발생 당시 트위터에서 빠르게 지진과 쓰나미, 원전 사고에 관해 전파된 사례가 있고, 2012년 10월 22일 허리케인 샌디가 발생했을 전후의 포스퀘어(Foursquare)의 체크인 데이터와 뉴욕의 피해 상태를 일치함을 살펴볼 수 있어 재난 피해 분석에서 소셜미디어 데이터가 연구 및 재난 분석에 유의미하게 사용되었다(Pu, 2013). 이처럼 재난 관리 분야에서 재난 예방 및 분석, 나아가 재난 사고를 예측하기 위해서 소셜미디어 데이터와 같은 비정형 데이터의 적절한 분석이 필요하다.

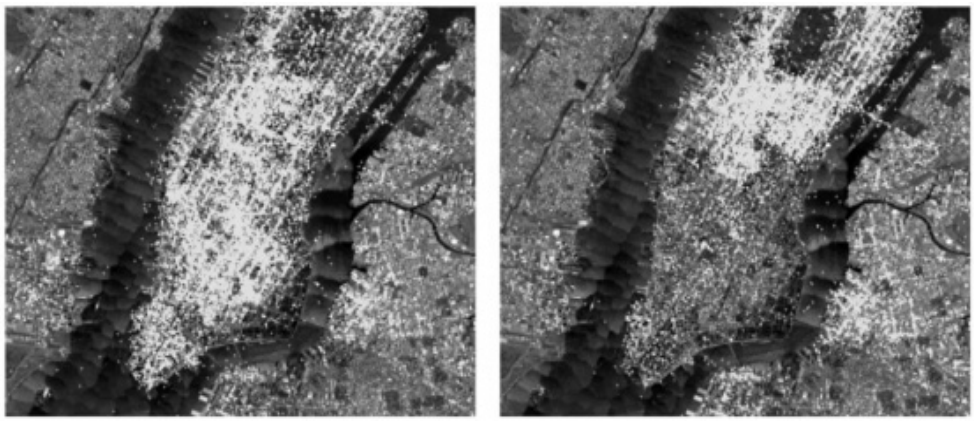


그림 1-1. 허리케인 샌디 발생 전후 체크인 데이터

트위터와 같은 소셜미디어 데이터 속에는 실제의 정보보다 훨씬 많은 양의 불필요한 데이터와 무의미한 데이터가 존재한다. 기존에는 이를 구분해내고 필요한 데이터만을 추출하기 위한 문서 분류 기법으로 패턴인식(pattern recognition), SVM(Support Vector Machine), LDA(Latent Dirichlet Allocation) 등과 같은 기계학습 방식을 사용해왔다.

2012년 이미지넷(Imagenet Large Scale Visual Recognition Challenge, ILSVRC)에서 토론토 대학이 84%의 정답률을 기록하여 압도적인 성능을 보임으로써 딥러닝(Deep learning) 기술이 대두되었다. 기존의 방법보다 10%가량 정확도를 향상한 것이었다. 기존의 기계학습 및 기타 알고리즘은 사람이 이미지나 텍스트에 대한 특징(feature)을 직접 설정하고, 최적화된 모델을 설계하는 구조였다. 하지만 딥러닝 알고리즘은 여러 개의 레이어를 겹쳐 쌓아 생긴 무수히 많은 파라미터를 컴퓨터가 직접 조정해가며 최적화된 해를 구한다. 딥러닝 알고리즘은 데이터가 많을수록 과적합(overfitting) 되지 않는 학습 결과를 보여주기 때문에 빅데이터 분석에 효과적이고 적합한 알고리즘으로 이미지 분석, 텍스트 마이닝 등 다양하게 사용되고 있다.

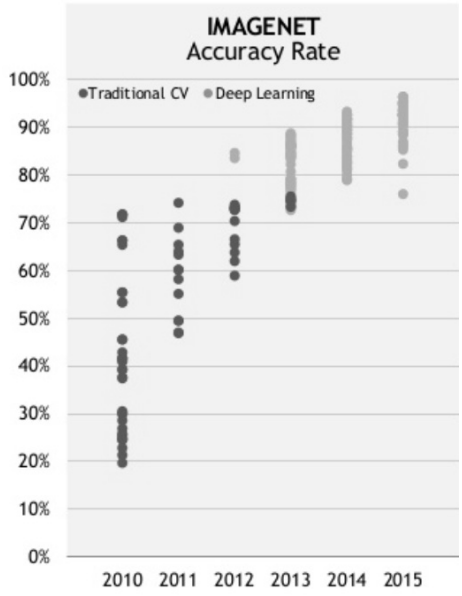


그림 1-2. 이미지넷 정답률(NVIDIA 2016 conference)

딥러닝 알고리즘 모델이 좋은 성능을 내기 위해서는 대용량의 데이터 세트가 필요하다. 따라서 대부분의 텍스트 마이닝 연구에서는 학습 데이터로써 대용량의 말뭉치(corpus)를 사용하고 있다. 하지만 데이터 생성과 이용에는 한계가 있어 많은 연구자가 데이터 증강 기법에 관해서 연구하여, 적절한 데이터의 증강은 모델 성능에 영향을 준다고 밝혀냈다(Zhang, 2015).

따라서 본 연구에서는 소셜미디어 데이터를 이용한 딥러닝 모델에 데이터 증강 기법을 적용하여 재난 정보 중 특히 화재 재난을 포함한 문장을 텍스트 데이터에서 탐지하는 정확도를 향상하는 것을 목적으로 한다. 제안된 모델은 위성 영상이나 센서로 습득된 데이터가 아니라 사람이 작성한 데이터에서 재난 정보를 습득할 수 있다는 점에서 의의가 있다. 이후, 습득된 재난 정보를 바탕으로 재난 이력을 관리하거나 실시간 재난 정보를 제공하는 등의 활용 방안도 제안하고자 한다. 또 딥러닝을 이용한 재난 탐지모델은 기존의 패턴인식과 같은 기계학습 알고리즘과 다르게 사용자의 개입이 적고 입력된 데이터를 통해서 모델이 파라미터를 조정하며 스스로 학습해 나간다는 장점이 있다.

1.2 연구 동향

재난 탐지 관련 연구는 주로 센서를 통해 획득한 데이터를 이용하여 이루어진다. 센서 데이터는 정형화되어있어 텍스트 데이터 분석보다 간단하고 빠른 분석이 가능하기 때문이다. 하지만 소셜미디어 데이터를 이용하면 사용자를 하나의 센서로 이용하여 재난 발생 이력을 관리하거나 과거 재난의 경향 분석 및 모니터링에 관한 정보도 수집할 수 있다는 장점이 있다.

텍스트 마이닝을 다양한 언어에 적용하기 위해서는 각 언어에 대한 학습 데이터가 필요하다(Ruder, 2016). 따라서 본 연구에서는 한국어 데이터에 대해 한정하여 실험하였다. 한글은 영어와 달리 조사가 존재하고 어미가 복잡한 특성이 있어 한글 텍스트를 분석할 때 어려움이 있다(장경애, 2015). 따라서 소셜미디어 데이터의 텍스트를 사용하기보다 네트워크 구조만을 분석하기도 하여(김재훈, 2017), 표 1-1과 같이 소셜미디어 데이터의 텍스트를 이용하여 재난 탐지 분석을 한 연구들을 위주로 연구 동향을 살펴보았다.

Earle(2012)은 트위터에서 지진과 관련된 단어인 “earthquake”, “gempa”, “temblor”, “terremoto”, “sismo”와 같은 키워드가 포함된 트위터를 불러온 후 유저의 위치 정보가 담겨 있는 트위터를 이용하여 지진을 감지하는 모델을 제안하였다. 이 모델은 실제 지진의 건수보다 적은 양의 지진을 탐지하지만 지진 발생 시간 2분 이내에 탐지하는 빠른 성능을 보여주었다.

조민희(2015)는 SNS, 뉴스, 커뮤니티 게시글, 정책데이터 등에서 재난 발생과 원인, 재난의 종류 등을 탐지하기 위해 개체명에 대한 사전을 작성하고 재난 정보를 탐지하였다.

Lazard(2015)는 에볼라 바이러스에 대한 질병 관리 센터와 주고받은 트윗 데이터를 수집한 후 ‘SAS Text Miner’를 통해 토픽 분석을 하여 사람들이 에볼라 바이러스에 대해 어떻게 생각하는지에 대해 분석하였다. 이는 사전에 지정한 데이터에 대해 제한적으로 실험을 수행하였기 때문에, 트위터 전체 데이터에 적용하기 어렵다.

임준엽(2015)은 트위터에서 지명 키워드를 이용하여 이벤트 종류와 상관없이 지명과 관련된 키워드가 증가하면 그 지역에 이벤트가 발생했다고 가

정하는 이벤트 위치 탐지모델을 제안하였다. 이는 어떠한 이벤트가 발생하면 사용자들이 그와 관련된 트위터들을 작성하는 경향을 이용한 것이다. 하지만 이벤트의 종류를 알 수 없고, 동음이의어에 대해서 구별할 수 없다는 한계가 있다.

국내의 재난안전연구원에서는 재난 예보 및 대책 수립에 사용하는 소셜 빅보드를 운영하고 있다. 이는 최선화(2016)의 소셜미디어 위험도 기반 재난이슈 탐지모델을 기반으로 한다. 최선화(2016)는 위험도 기반 전조 이슈 탐지모델과 어휘패턴 기반 재난 발생이슈 탐지모델 두 가지를 제안하였는데, 이 중 어휘패턴 기반 방식의 경우에는 '화재'가 '휩쓸다'라는 단어의 5글자 이내에 있으면 화재라고 판단한다. 해당 연구에서 화재에 대한 탐지 정확도는 51%(8,191/16,074)로 낮은 편이며 사전이나 키워드 이용방식과 같이 연구자가 직접 모든 패턴을 입력해야 한다는 단점이 있다.

하현수(2016)는 트위터의 위치를 확정하기 위한 지명 단어의 노이즈 제거 기법과 랜드마크를 이용하여 지명 단어를 확정하는 기법을 제안하였다. 노이즈 제거 기법은 불용어 사전을 만든 후 지명 단어의 모든 사례를 직접 작성하고 사전을 갱신하여 노이즈를 제거한다. 지명 확정 기법은 랜드마크에 관한 사전을 이용하여 사전에 해당하는 키워드를 가진 트위터에 랜드마크의 위치를 부여한다.

Burel(2017)은 BOW(bag of words)를 이용하여 문장을 학습하는 CNN 모델과 특정 키워드에 대해서만 문장을 학습하는 CNN 모델을 융합한 Dual-CNN 모델을 이용하여 트위터에서의 이벤트를 탐지하는 모델을 제안하였다. 홍수와 같은 재난에서는 79% 이상의 정확도를 보였지만 그 외에는 SVM 모델과 비슷하거나 조금 나은 성능을 보였다.

유호선(2018)은 대형 재난이슈를 대상으로 해당 재난에 대한 키워드로 트윗, 뉴스를 수집하여 사회 재난들의 생존 주기를 연구하였다. 생존 기간을 해당 이슈가 트위터 상에서 언급이 되는 기간으로 잡고, 생존 주기의 유형에 따라 재난을 분류하였다. 분류된 재난들에 대하여 재해 통계 연감에 있는 지표와 기간을 비교하여 의사 결정 나무(Decision Tree)를 통해 관련이 큰 지표들을 추출하였다.

이처럼 국외에서는 Burel(2017)의 연구와 같이 딥러닝을 사용하여 재난 정보 문장탐지 모델을 사용하였지만, 한국어에서는 언어의 특성이 영어와 달라 모델을 비교할 수 없다. 또, Burel(2017)의 연구는 정확도가 기존 기계학습 방식에

비해 크게 향상되지 않았다는 단점이 있었기 때문에 본 연구에서는 텍스트 분석에서 더 향상된 성능을 보여주고 있는 순환신경망 기법을 적용하였다. 신동원(2017)에 따르면 이 모델은 한국어 데이터에서도 SVM, CNN과 비교하여 0.6% 향상된 정확도를 보였다.

반면 국내에서 진행된 대부분의 연구는 사용자가 사전을 만들고 그에 해당하는 키워드를 찾거나 패턴과 일치하는지 확인하여 재난 정보 문장 여부를 판별하였다. 이는 재난 정보에 해당하는 키워드나 패턴을 비롯하여 모든 예외 처리에 대한 정보까지 사용자가 직접 사전을 추가해야 해서 오랜 기간이 필요하다는 단점이 있다. 최선화(2016)의 탐지모델은 2012년부터 꾸준히 계속 사전을 업데이트하여 실제로 국립재난안전연구원이 운영하는 소셜 빅보드 시스템에서도 활용 중이다.

따라서 본 연구에서는 딥러닝 중 텍스트 분석에서 성능이 좋은 순환 신경망 알고리즘을 이용하고, 학습된 Word2Vec 모델을 통해 데이터를 증강하여 사용자의 간섭을 최대한 줄이고 더욱 정확한 재난 정보가 담긴 문장을 탐지하는 모델을 제안하여 최선화(2016)의 모델과 비교하여 정확도를 평가해보았다. 이때 순환신경망 모델에서는 LSTM과 GRU의 성능이 데이터의 종류에 따라 다르게 나왔다는 조휘열(2016)의 연구에 따라 두 가지 방식을 적용하여 평가하고, 그에 따라 데이터 증강 기법이 결과에 유의미한 영향을 주었는지에 대해 분석해보았다.

표 1-1. 재난 탐지 연구 동향

저자	연구주제	연구내용		
		사전 기반	기계학습	딥러닝
Earle (2012)	Twitter earthquake detection: earthquake monitoring in a social world	●		
조민희 (2015)	재난 이벤트탐지를 위한 지식베이스 구축	●		
Lazard (2015)	Detecting themes of public concern: A text mining analysis of the Centers for Disease Control and Prevention's Ebola live Twitter chat	●	●	
임준엽 (2015)	트위터 기반의 실시간 이벤트 지역 탐지 시스템	●		
최선화 (2016)	소셜 미디어 위험도 기반 재난이슈 탐지모델	●	●	
하현수 (2016)	트위터를 활용한 실시간 이벤트탐지에서의 재난 키워드 필터링과 지명 검출 기법	●		
Burel (2017)	On semantics and deep learning for event detection in crisis situations			●
유호선 (2018)	재난 사건별 이슈 생존 주기 유형 분석	●	●	

1.3 연구 범위 및 방법

사회적으로 이슈가 되는 대형 재난들은 사람들이 SNS에 공유하여 빠르고 널리 퍼지게 된다. 하지만 SNS에서는 불필요한 데이터의 생성이 같이 일어나기 때문에 현상 분석을 위해 적합한 데이터를 판별하는 것이 중요하다. 비정형 데이터인 SNS 텍스트 데이터에서 특정 재난 정보를 정확히 탐지해 낼 수 있다면 탐지된 SNS 데이터를 이용한 분석의 정확도도 더 높아지고, 재난 발생 위치 추정 및 예측 등의 다양한 활용이 가능하다.

본 연구에서 사용하는 데이터는 RNN 모델의 학습을 위한 뉴스 데이터와 평가를 위한 트위터 데이터, 그리고 Word2Vec 모델의 학습을 위한 위키피디아 데이터가 있다. 트위터 데이터에서는 재난 관련 문장이 적기 때문에 학습에 적합하지 않아 “화재” 키워드로 검색한 뉴스 데이터를 학습 데이터로 사용하였다.

딥러닝 모델은 학습 데이터(training data)가 적으면 과적합이 일어나 모델의 성능이 매우 떨어지게 된다. 하지만 재난 문장을 직접 분류 후 라벨링(labeling) 하므로 만들 수 있는 학습 데이터의 양에 한계가 있었다. 충분한 학습 데이터를 확보하기 위해 데이터 증강(augmentation)이 필요하다고 판단하여 데이터 증강을 위한 Word2Vec 모델을 만들었다. 실험의 자세한 내용은 3. 실험 방법에서 설명하였다.

본 연구에서 비정형 데이터를 이용한 재난 문장탐지 방법의 흐름은 그림 1-3과 같다.

첫 번째 단계는 데이터를 증강하고자 Word2Vec 모델을 학습시키는 과정이다. Word2Vec 모델은 단어들을 벡터화하여 단어들 사이의 관계를 벡터로 표현할 수 있게 하는 가장 효과적인 워드 임베딩 기법의 하나이다. Word2Vec 모델은 학습된 데이터가 크면 클수록 더 정확한 단어 관계를 나타낼 수 있으므로 한국어 위키피디아의 전체 문서를 이용하여 학습하였다. 데이터 증강은 Word2Vec 모델을 이용해 학습 데이터의 문장에 있는 모든 명사를 동의어로 치환하는 방식으로 진행하였다.

두 번째 단계에서는 딥러닝 모델(LSTM, GRU)에 학습 데이터를 입력하여 모델의 각 파라미터를 학습시키는 과정이다. 학습이 완료된 후에 정확도 평가를

하였다. 본 연구에서는 최선화(2016)에서 사용했던 패턴인식 기법의 결과를 F1-measure를 이용하여 비교해 보았다.

마지막 단계로는 가장 학습이 잘된 모델을 이용하여 트위터 데이터에서 재난 문장탐지 모델을 적용해보았다. 또, 본 연구의 재난 문장탐지 모델을 이용하여 어떤 분야에 활용할 수 있을지 제안해보았다.

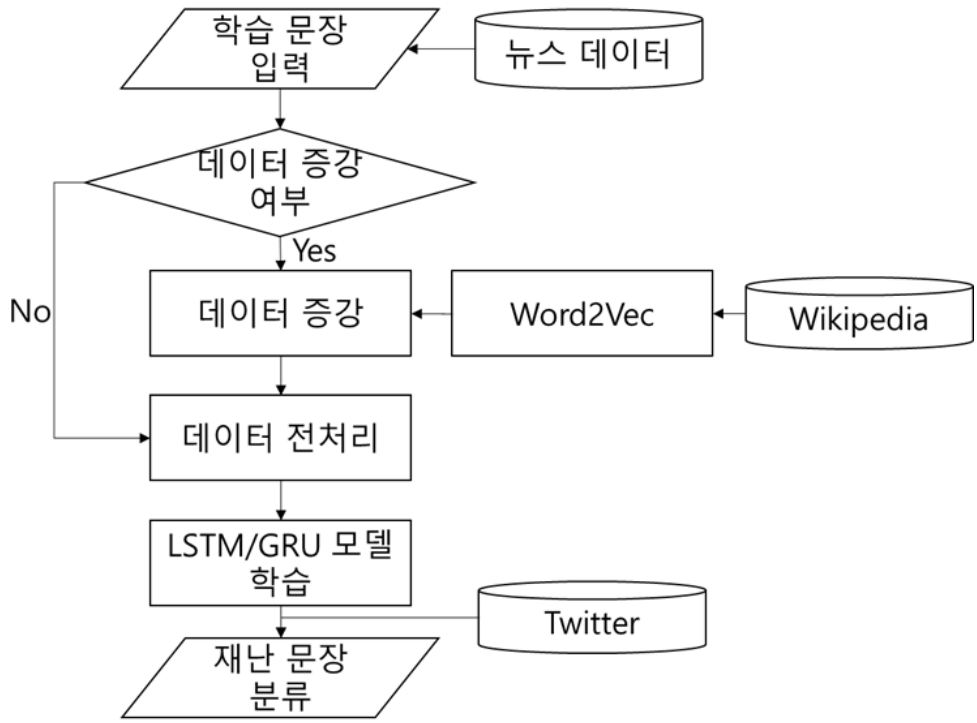


그림 1-3. 연구 흐름도

2. 이론적 배경

2.1 텍스트 전처리

2.1.1 워드 임베딩 기법

워드 임베딩(Word Embedding)이란 원시 형태의 텍스트로 이루어진 데이터를 딥러닝 및 기계학습 알고리즘에 적용하기 위해 하나의 단어를 벡터 공간상의 하나의 점으로 치환하여 숫자 형식으로 변환하는 것을 말한다. 즉, 텍스트 데이터를 숫자로 변환하는 과정이며 기존의 연구에서는 One-hot encoding 방식을 널리 사용해왔다. 이 방식은 n개의 단어가 있을 때 길이가 n인 벡터를 만들어, 어떤 단어가 해당하는 자리에 1을 넣고 나머지 자리에는 0을 넣는 방식으로 사용한다. 이 방법은 단어가 많아지면 계산량이 급격히 늘어나며 단어 사이의 관계를 나타낼 수 없다는 큰 단점이 존재했다. 하지만 딥러닝의 성능 향상으로 신경망 모델을 이용하여 각각의 어휘의 관계를 벡터화한 NNLM(Neural Network Language Model)을 제안한 이후(Bengio, 2003), CBOW과 Skip-gram 방식을 도입한 신경망 모델인 Word2Vec를 제안하여 단어 의미 유사도 평가에서 Bengio의 연구보다 더 향상된 결과를 보여줬다(Mikolov et al. 2013). Word2Vec 모델은 기존의 방법에 비해 계산량을 줄여 빠른 학습을 가능하게 하여 워드 임베딩 기법 중 가장 대표적인 모델로 사용되고 있다.

Word2Vec에서 임베딩하려는 단어의 수를 V, 사용자가 지정한 은닉층(hidden layer)의 노드 수를 N이라고 하면, 모델은 크기가 $V \times N$인 가중치 행렬 W을 업데이트하면서 가장 최적화된 값을 찾는 방향으로 학습한다.

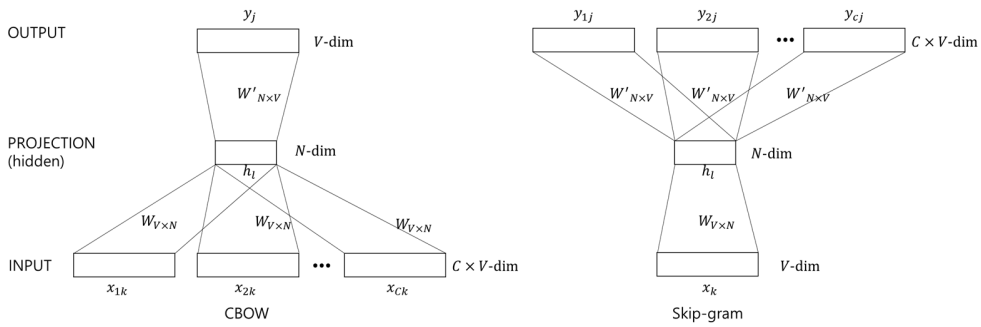


그림 2-1. CBOW, Skip-gram 모식도

그림 2-1처럼 CBOW는 주변에 분포하는 단어를 줬을 때($x_{1k}, x_{2k}, \cdots, x_{ck}$) 중심 단어($y_j$)를 예측하여 실제 값과의 차이의 손실을 줄이는 방향으로 학습을 진행하며, Skip-gram은 중심 단어를 줬을 때(x_k) 그 주변에 분포하는 단어($y_{1j}, y_{2j}, \cdots, y_{cj}$)를 예측하여 실제 값과의 차이의 손실을 줄이는 방향으로 학습이 진행된다. Mikolov의 연구에서는 데이터가 많을 때 Skip-gram에서의 단어 의미 유사도 평가의 정확도가 더 높게 나왔으므로 본 논문에서는 데이터 증강을 위해 Word2Vec을 적용할 때에는 Skip-gram 방식을 적용하였다.

Skip-gram에서 $w_1, w_2, w_3, \ldots, w_T$개의 단어를 임베딩 할 때, Word2Vec 모델은 식 2-1과 같이 로그 확률의 평균을 최대화하는 방향으로 학습해 나간다.

$$\frac{1}{T}\sum_{t=1}^{T}\left[\sum_{j=-k}^{k}\log p(w_{t+j}|w_t)\right] \tag{2-1}$$

k는 kernel window로써, 중심 단어에서 얼마나 떨어져 있는 단어를 볼 것인지를 나타내는 지표이다. k가 클수록 학습시간은 더 오래 걸리지만, 더 높은 유사도를 보일 수 있다(Mikolov, 2013).

2.1.2 데이터 증강 기법

데이터 증강 기법(Data Augmentation)이란 딥러닝 및 기계학습 분야에서 인위적으로 데이터의 양을 증가시키는 기술이다(Wan *et al.* 2014). 본래는 통계학에서 결측치(Missing value)를 처리하기 위한 기법으로 사용됐고, 딥러닝 분야에서는 주로 이미지 데이터에 적용되어 사용한다. 이미지의 레이블은 그대로 둔 채 이미지의 픽셀을 변화시켜 추가적인 데이터를 생성한다. 추가로 생성된 데이터는 알고리즘의 과적합을 방지하는 용도로 학습된다. 주로 이미지 데이터에서 사용되는 데이터 증강 방법으로는 Horizontal flips, Random crops/scales, Color jitter, Rotation 등이 있다. Flip은 이미지를 반전시키는 것이고, Crops/Scale은 알아볼 수 있는 내에서 이미지를 잘라 내거나 크기를 조절하는 것이고, Jitter는 이미지의 색상을 변경하는 것이다. 이처럼 생성된 이미지 데이터는 사람에게는 같게 인식되지만, 컴퓨터는 각각 새로운 이미지 데이터로 인식하여 학습할 수 있게 된다.

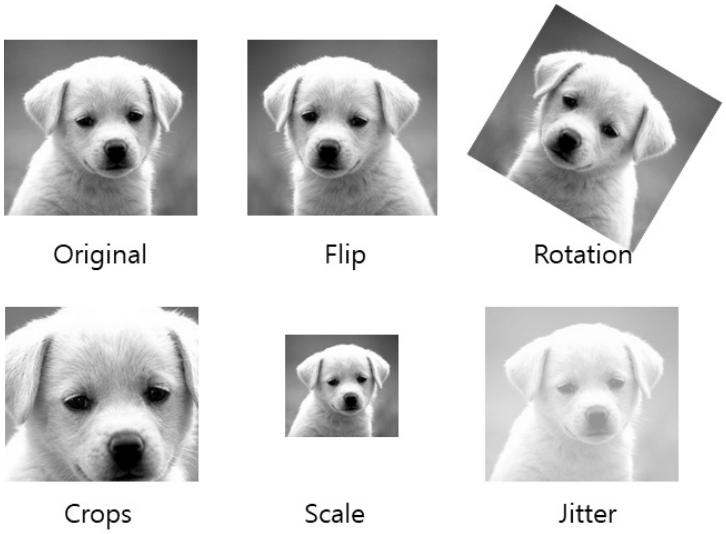


그림 2-2. 이미지 데이터 증강 예시

반면 텍스트 데이터에서는 문장 속 단어의 순서나 특정 표현이 빠짐으로써 문장의 의미가 달라질 수 있으므로 위와 같은 증강 기법을 사용할 수 없다. 이에 영어와 관련하여 Zhang(2015)은 Thesaurus의 유의어 사전을 통한 문장 속 단어를 유의어로 대체하여 생성된 유의어 문장을 추가로 활용하는 데이터 증강 기법을 제시하였다. 이는 사전 생성된 막대한 양의 유의어 사전이 필요하다는 단점이 존재한다. 또한, 국내에서 조휘열(2017)은 Seq2Seq모델을 통해 딥러닝을 통해 컴퓨터가 학습할 문장을 직접 생성하는 기법을 제시하였고, 모든 모델에서 Augmented 된 데이터가 학습 성능이 향상되었다는 결과를 보였다(조휘열, 2017). 하지만 이처럼 자동 생성된 문장에서는 필요한 단어가 들어가 있지 않거나 전혀 뜻이 다른 문장을 생성하는 오류가 존재한다는 단점이 있다.

2018년 3월 20일 Kaggle의 “Toxic Comment Classification Challenge”에서 영어로 이루어진 데이터를 다른 언어로 번역 후, 다시 영어로 번역한 문장을 학습 문장으로 활용하여 2등을 기록하는 등(Neongen, 2018) 아직 텍스트에서 데이터 증강 기법에 관한 연구는 활발히 연구 중이다. 아직 한글에서는 텍스트 데이터 증강 기법에 대한 연구결과로 나타난 기준이 명확하지 않다.

본 논문에서는 앞서 기술한 Word2Vec을 이용하여 한글 위키피디아의 모든 문서에 대하여 Word2Vec 모델을 학습하고, 학습된 모델을 이용하여 유의어로 대체하는 데이터 증강 방식을 적용하였다. 이는 사용된 언어와 상관없이 Word2Vec 모델이 학습된 결과를 유의어로 제시하게 되며 사전 작성된 유의어 사전이 필요 없다는 장점이 존재한다.

2.2 순환 신경망

2.2.1 RNN(Recurrent Neural Network)

RNN(Recurrent Neural Network)은 인공신경망의 한 종류로서, 이미지 분석에 주로 사용되는 CNN(Convolution Neural Network)과는 달리 음성, 문자, 주가 등 시간에 따라 배열된(Sequence) 데이터 처리에 적합한 모델로 알려져 있다. RNN의 기존 인공신경망들과 가장 큰 차이점은 현재의 출력 결과가 이전 단계에 대한 메모리 정보로 저장되어 같은 입력 값을 넣더라도 이전에 들어온 입력값들에 따라 다른 출력 값이 나온다는 점이다. 즉, 피드백을 통해 이전까지의 입력 값들에 대한 정보를 기억하고 다음 단계로 전달하기 때문에 순차적인 데이터 처리에 강점을 나타내고 있다.

RNN은 그림 2-3과 같이 다양한 방식으로 연결하여 입력값과 출력값의 처리가 가능하여 사진에 단어들의 나열을 붙여 설명을 만들거나, 단어들의 나열을 분석하여 특정한 값을 추출하여 문장에 대한 감정 분석, 분류 등을 하기도 한다. 또 텍스트 데이터에서 가장 많이 연구되는 분야인 단어의 나열을 다른 언어로 번역을 하는 등의 다양한 방법에 적용할 수 있다.

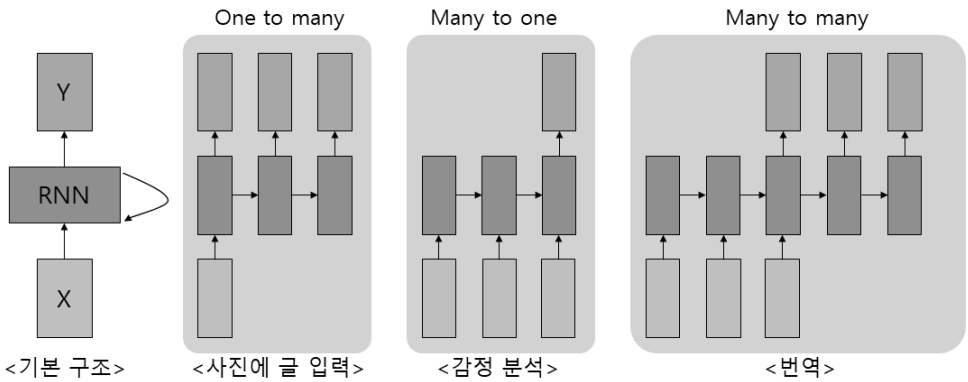


그림 2-3. RNN의 활용 예시

Elman(1990)의 가장 기본적인 RNN 모델은 그림 2-4와 같이 체인 구조로 구성되어 있으며 각 기본 구조와 state를 펼쳐 놓을 때를 나타내었다.

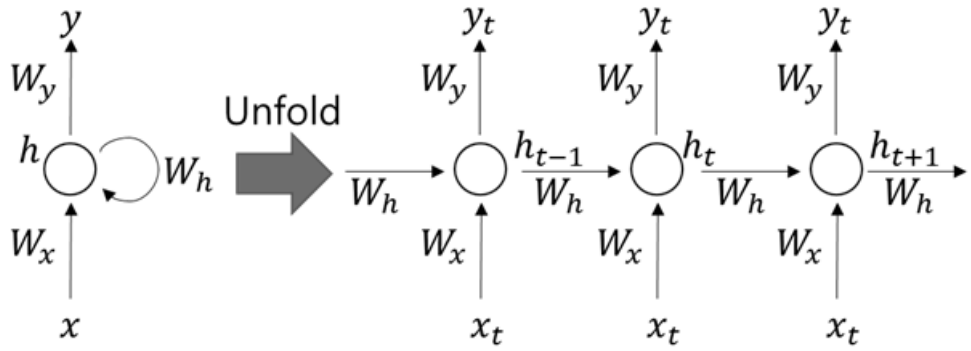


그림 2-4. 기본적인 RNN 모식도

$t-1$에 갱신된 h_{t-1}이 t단계에 반영되며 h_t의 인자를 수식으로 나타내면 식 2-2와 같다.

$$h_t = f_W(h_{t-1}, x_t) \tag{2-2}$$

h_t는 t에서의 hidden state를 나타내고, x_t는 t에서의 입력값이며 W는 가중치 행렬이다. f_W는 활성 함수(activation function)로 보통 tanh와 같은 시그모이드 함수를 사용하여 식 2-3과 같이 표현할 수 있다. hidden state는 $t-1$의 정보와 t의 입력값을 받아 갱신된 후, 출력값 y_t로 식 2-4와 같이 나타난다.

$$h_t = \tanh(W_h h_{t-1} + W_x x_t) \tag{2-3}$$

$$y_t = W_y h_t \tag{2-4}$$

RNN 모델은 순전파(forward propagation)와 역전파(backpropagation)를 통해 경사 하강법(gradient descent)으로 가중치 값들을 갱신하며 학습을 진행한다.

Elman의 모델은 역전파 시에 tanh함수가 양쪽 끝에서 기울기 값이 0으로 수렴하기 때문에 기울기 손실(vanishing gradients)이 발생한다. 이로 인해 장기 의존성(Long-term Dependency)의 문제가 발생하여 먼 정보일수록 잘 잊어버린다는 단점이 존재한다(Bengio, 1994).

경사 하강법이란 딥러닝 네트워크의 예측 결괏값과 실제 결괏값의 차이를 정의하는 loss function의 $J(\theta)$의 값을 최소화하기 위해, 기울기를 이용하는 방법이다. loss를 구할 때 cross-entropy 함수는 식 2-5와 같다.

$$J(\theta) = -\sum P(y|x)\log P(y|x;\theta) \tag{2-5}$$

$P(y|x)$는 실제값을 의미하고 $P(y|x;\theta)$은 예측값을 나타낸다. 한 state에서의 기울기 변화 식은 식 2-6과 같다.

$$\theta = \theta - \alpha \nabla_{\theta} J(\theta) \tag{2-6}$$

α는 learning rate로 학습이 진행되는 속도를 나타내며 사용자가 조정하는 하이퍼 매개변수(Hyperparameter)에 속한다. 경사 하강법을 진행할 때, 전체 학습 데이터를 사용하는 Batch Gradient Descent는 막대한 계산량이 필요하다. 따라서 데이터를 mini-batch로 나누어 적용하는 Mini-batch gradient descent(MGD)나 일부 데이터만 잘라내어 적용하는 Stochastic Gradient Descent(SGD) 방법이 알려져 있다. 본 연구에서는 MGD 방식을 적용하였다.

2.2.2 LSTM(Long Short-Term Memory Unit)

앞서 언급한 RNN의 문제점을 해결하기 위해 Hochreiter(1997)은 cell-state를 추가한 LSTM(Long Short-Term Memory Units) 모델을 제안했다. LSTM은 기본적으로 Elman의 RNN과 같은 구조로 이루어져 있으나 각 state에 게이트(gate)들을 추가하여 전달할 정보들을 선택하고 셀 스테이트(cell-state)를 통해 결괏값을 갱신하여 전달한다. LSTM의 구조는 그림 2-5와 같다.

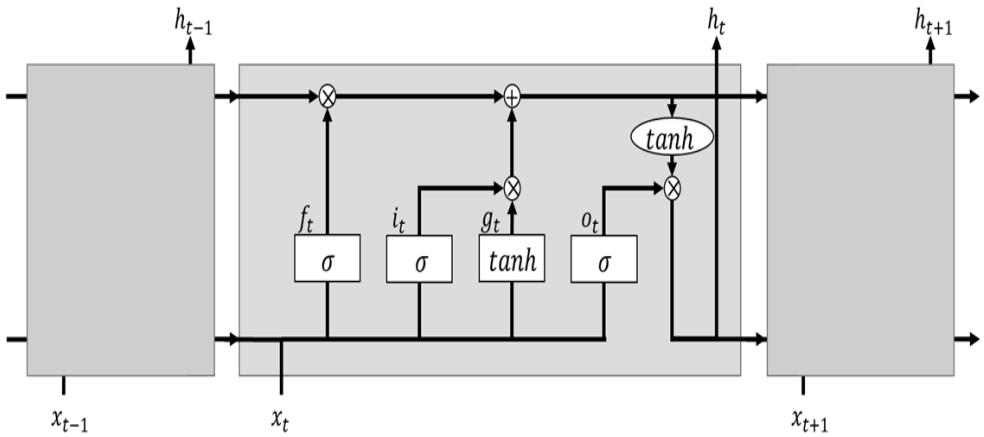


그림 2-5. LSTM의 구조

LSTM 모델에서 새로 추가된 게이트들은 망각(Forget), 입력(Input), 출력(Output) 게이트라고 하며, LSTM 한 셀의 흐름은 그림 2-6, 그림 2-7, 그림 2-8, 그림 2-9와 같다(Kapathy, 2016).

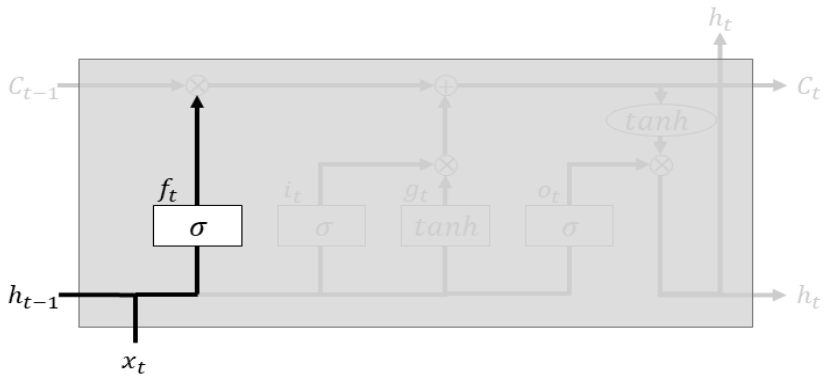


그림 2-6. LSTM-1

$$f_t = \sigma(W_h^f h_{t-1} + W_x^f x_t) \tag{2-7}$$

σ는 시그모이드 함수를 뜻하고 0에서 1 사이의 값을 출력한다. 먼저, 그림 2-5의 망각 게이트(식 2-7)에서는 잊어버릴 정보를 선택하는 과정으로, h_{t-1}, x_t를 입력값으로 받은 후, 시그모이드 함수를 통해 요소(element)를 기억할 것인지 지워버릴 것인지를 결정한다.

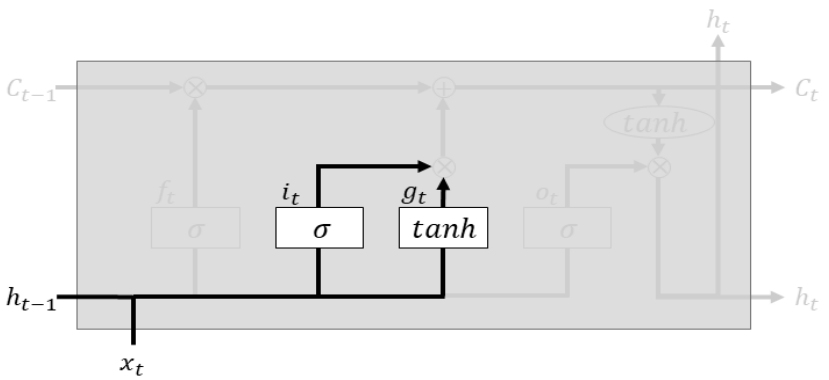


그림 2-7. LSTM-2

$$i_t = \sigma(W_h^i h_{t-1} + W_h^i x_t) \tag{2-8}$$

$$g_t = \tanh(W_h^g h_{t-1} + W_h^g x_t) \tag{2-9}$$

다음 단계로 그림 2-7과 같이 새로운 정보를 셀 스테이트에 저장할지를 두 단계를 통해 결정한다. 입력 게이트(식 2-8)에서 새로운 입력값을 갱신할 것인지를 결정하고, 기본 RNN 모델의 hidden state와 같이(식 2-9) tanh 함수를 이용하여 t에서의 -1부터 1 사이의 입력값을 만들어낸다.

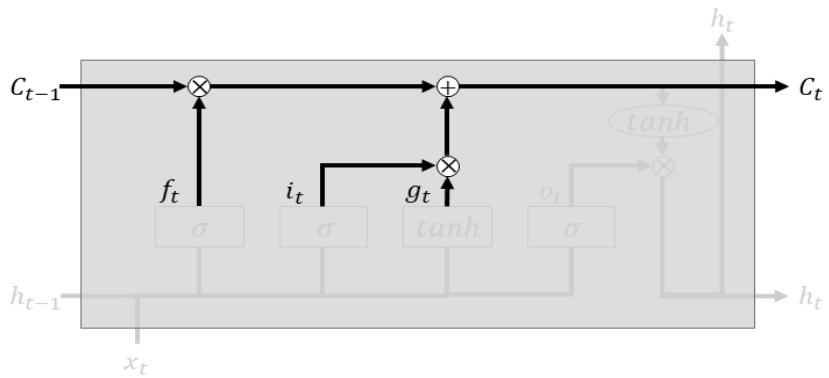


그림 2-8. LSTM-3

$$c_t = f_t \otimes c_{t-1} + i_t \otimes g_t \tag{2-10}$$

⊗는 요소별(element-wise) 곱셈을 뜻한다. 식 2-10과 그림 2-8에서는 $t-1$의 셀 스테이트에서 불필요한 요소를 망각 게이트의 결괏값과 곱하여 삭제한다. 그 후, t에서 새로운 입력값을 입력 게이트의 결괏값과 곱하여 더하여 t에서의 셀 스테이트 값을 갱신한다.

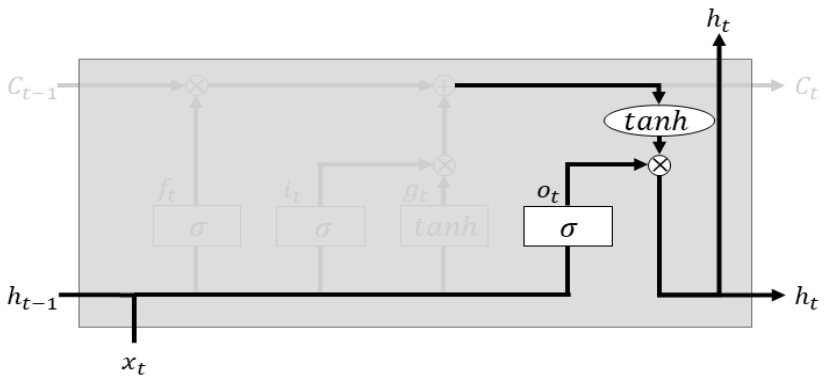


그림 2-9. LSTM-4

$$o_t = \sigma(W_h^o h_{t-1} + W_h^o x_t) \tag{2-11}$$

$$h_t = o_t \otimes \tanh(c_t) \tag{2-12}$$

마지막 단계인 그림 2-9의 구조와 같이 출력 게이트(식 2-11)를 통해 출력될 요소들이 선택된 후 식 2-11과 같이 t에서의 셀 스테이트의 정보를 곱하여 원하는 값만 현재 정보인 h_t 값으로 갱신한다. LSTM 모델에서도 일반적인 RNN과 마찬가지로 경사 하강법을 통해 역전파를 계산하여 가중치를 학습한다 (Graves, 2012).

LSTM은 셀 스테이트에 정보를 입력, 출력, 망각 게이트를 통해 담음으로써 중요한 정보는 더 오래 기억하고, 불필요한 정보는 삭제하며 RNN에서의 기울기 손실 문제를 완화하여 성능을 크게 향상했다. 이후 LSTM의 구조를 변형한 연구들이 발표되었고 Greff(2017)은 각 변형 알고리즘의 성능을 TIMIT 데이터 세트[1]에서 평가하여 각 모델의 구조보다는 learning rate와 같은 하이퍼 매개변수가 성능에 가장 큰 영향을 준다고 하였다.

1) TIMIT은 미국 영어 사용자들의 성별과 다양한 방언에 따라 모아놓은 음성 말뭉치(corpus)이다. 자동 음성 인식 시스템을 위해 설계되었다(Wikipedia).

2.2.3 GRU(Gated Recurrent Unit)

LSTM의 많은 변형 중 조경현(2014)의 GRU(Gated Recurrent Unit) 모델은 LSTM의 별도로 셀 스테이트를 두지 않고 학습을 진행하기 때문에, 필요한 매개변수 수가 LSTM보다 적어 학습 속도가 매우 빠르고 데이터의 수도 적게 필요로 한다. GRU 모델의 구조는 다음과 같다.

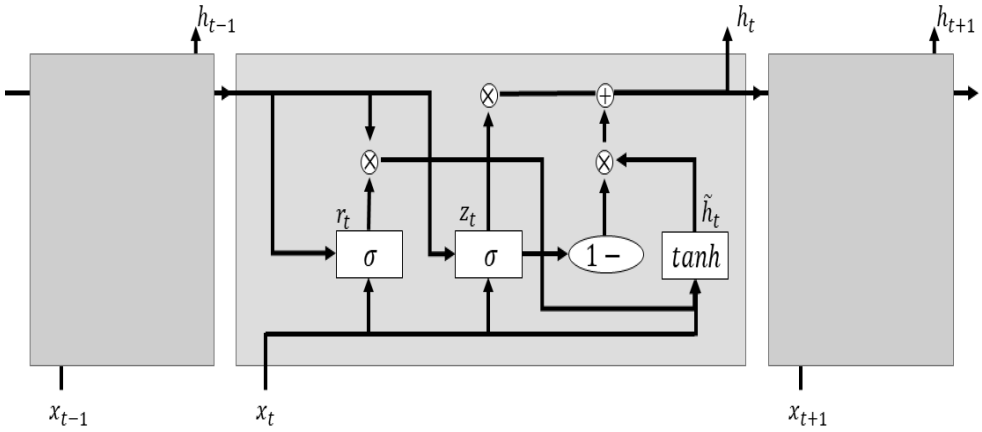


그림 2-10. GRU의 구조

GRU는 LSTM 모델의 입력, 망각 게이트를 합친 업데이트 게이트(update gate, 식 2-11)와 리셋 게이트(reset gate, 식 2-12)를 제안했다. 전체적인 흐름은 다음과 같다.

$$z_t = \sigma(W_h^z h_{t-1} + W_x^z x_t) \tag{2-11}$$

$$r_t = \sigma(W_h^r h_{t-1} + W_x^r x_t) \tag{2-12}$$

리셋 게이트와 업데이트 게이트는 LSTM의 게이트들과 같은 원리로 0부터 1 사이의 값을 출력한다.

GRU 셀의 흐름은 그림 2-11, 2-12와 같이 나타낼 수 있다.

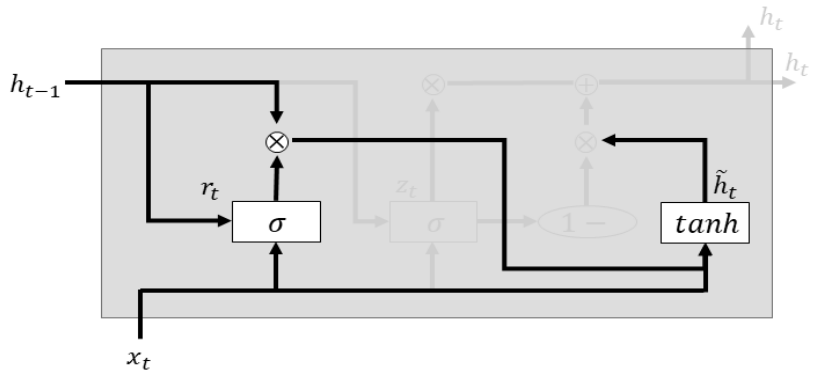


그림 2-11. GRU-1

$$\widetilde{h}_t = \tanh(W_{hh}(r_t \otimes h_{t-1}) + W_{xh}x_t) \tag{2-13}$$

그림 2-11은 $\widetilde{h}_t$에 정보를 저장하는 과정을 나타내었다. 식 2-13은 $\widetilde{h}_t$에 정보를 저장할 때, 리셋 게이트 r_t를 이용하여 $t-1$의 정보 h_{t-1}를 선택하여 저장한다는 뜻이다. 리셋 게이트의 값이 1이면 과거 정보를 모두 저장하고, 0이면 모두 잊는다. 이때, 리셋 게이트의 값과 상관없이 t에서의 정보 x_t는 모두 저장한다.

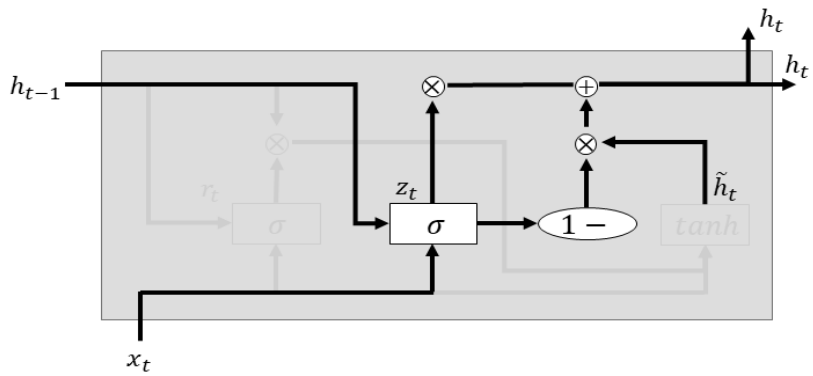


그림 2-12. GRU-2

$$h_t = z_t \otimes h_{t-1} + (1 - z_t) \otimes \widetilde{h}_t \tag{2-14}$$

그림 2-12는 조합된 현재 정보인 $\widetilde{h_t}$과 과거 정보인 h_{t-1}을 업데이트 게이트 z_t를 이용하여 h_t로 갱신한다. 식 2-1은 그 과정을 수식으로 나타내었다. 업데이트 게이트가 0이라면 과거 정보는 모두 잊고 현재는 모두 기억하고, 1이라면 과거는 모두 기억하고 현재 정보는 잊는다.

GRU는 LSTM보다 훨씬 빠른 학습 속도를 가지고 기존의 알고리즘과 비슷하거나 더 좋은 정확도를 보여주고 있다(Chung, 2014). 보통 GRU보다 LSTM이 대용량 데이터에서 더 좋은 성능을 보인다고 알려졌지만, 조휘열(2016)은 대용량 텍스트 분류 기술에 적용할 때 GRU가 더 높은 정확도를 보인다고 발표했다.

3. 실험 방법

3.1 데이터

실험 학습에는 “화재, 불”의 키워드로 검색된 2017년 1월부터 2018년 1월까지의 조선일보 뉴스 기사 903개를 모델의 학습에 적용하고, 선행연구와의 정확도를 비교해 보았다. 2017년 12월 14일부터 12일 26일까지 수집된 트위터 364,477개에 학습된 모델을 적용하여 평가하였다. 데이터 증강을 위한 Word2Vec 모델 학습에는 한국어 위키피디아에 등록된 문서 416,391개[2]개를 이용하였고, 모델의 학습에는 조선일보 뉴스 기사 903개 중 ‘Text’ 항목을 불러와 21,041개의 문장으로 나누어 그중 5,931개의 문장을 라벨링 하여 적용하였다. 화재 관련 문장 분류의 기준은 다음 표 3-1과 같다.

표 3-1. 문장 분류 기준

Class	분류 기준
0	재난 발생과 관련 없는 문장
1	화재 어휘패턴에 포함되는 문장

화재 재난 발생 여부의 정보 판단 여부는 선행연구인 박성공(2015)의 “재난 빅데이터 기반 전조 감지 기술개발 및 소셜 빅보드 확산전략 수립”의 “화재 재난 어휘패턴 규칙”을 참고하여 직접 분류하였다. 표 3-2과 표 3-3은 재난 어휘패턴 규칙과 분류된 학습 데이터 5,931개 중 일부만을 나타내었다.

2) https://ko.wikipedia.org/wiki/한국어_위키백과, (접속일: 2018년 02월 13일)

표 3-2. “화재” 재난에 대한 어휘패턴 규칙(박성공, 2015)

패턴 종류	패턴
포함조건	([[:시설물:]]|[[:이동수단:]])\w{0,5}(불|화재)
	(화재|불길|화염)\w{0,5}(현장|발생|[[:휩쓸다:]]|[[:일다:]]|치솟)
	(불|화재)\w{0,5}(사상|사망|부상|화상|환자|이송|인명|피해|대피|재산|발생)
배제조건	[[:게임:]]|책|소설|저서|저자|지은이
	(불펜|야구|투수|작가)\w{0,5}(불길|불질)금융\w{0,5}불길
	오늘의\s{0,1 }역사|엑소|exo|백현|마블
	(화재|불|불길|화염)\w{0,5}(교육|도시형생활주택|드라이비트|살해|스마우그|안전관리|안전점검|압수수색|예방|원인|이유|조사|타살|학구열|현장감식|훈련|건의|대통령|무능|종편|왜|가능성|뉴스\s{0,1}예고|상식|역사|특별재난지역|선포|개그콘서트|요령|주의|소실|정전\s {0,1}피해|재산\s{0,1}피해|천만\s{0,1}원|감식|발화점|여원|씨랜드|숭례문|세월호)
	(세월호|숭례문|씨랜드|여원|발화점|감식|천만\s{0,1}원|정전\s{0,1}피해|재산\s{0,1}피해|소실|주의|요령|개그콘서트|특별재난지역|선포|역사|상식|뉴스\s{0,1}예고|가능성|왜|종편|교육|건의|대통령|무능|도시형생활주택|드라이비트|살해|살해하다|살해한|스마우그|안전관리|안전점검|압수수색|예방|원인|이유|조사|타살|학구열|현장감식|훈련)\w{0,5}(화재|불|불길|화염)

표 3-3. 학습 데이터 일부

Text	Class
불법 주차는 화재 때마다 피해를 키운 원인 중 하나로 지적된다	1
스포츠센터 화재 이후로 출동 사이렌이 울릴 때마다 가슴이 두근거리고 식은땀이 납니다	1
6일 오후 경북 상주에서 발생한 산불이 임야 13ha를 태우고 20여 시간 만에 진화됐다	1
⋮	⋮
정부는 건설현장에서 사고가 발생하면 건설공사 입찰자격 사전심사(PQ) 제도에서 불이익을 주는 환산재해율을 시행하고 있다.	0
전국에서 매일 사고 희생자가 발생하는데 정부는 매일 묵념해야 하나	0

3.2 설계

3.2.1 전체 구조

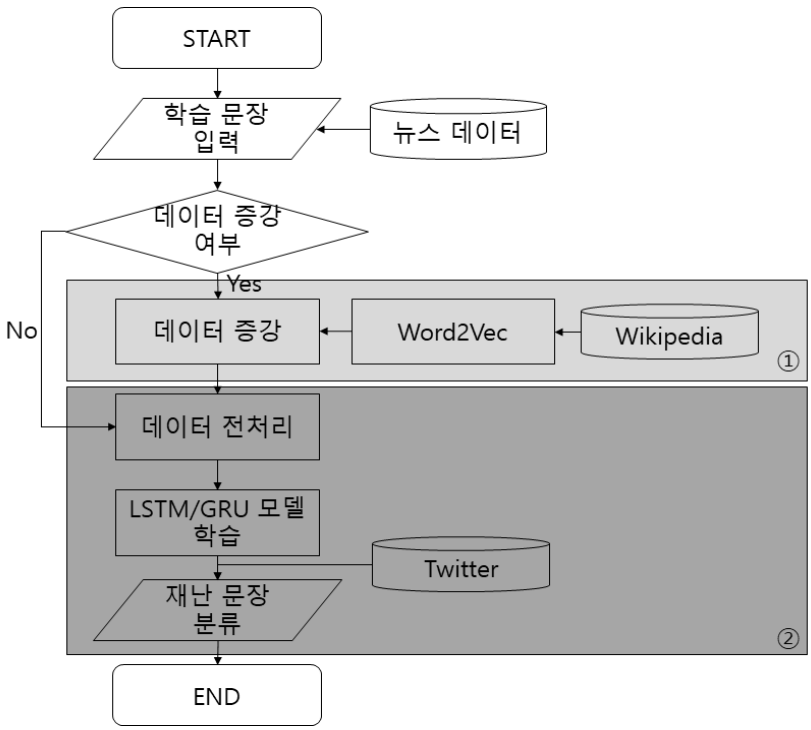


그림 3-1. 전체 구조

전체 재난 문장탐지 모델의 단계는 그림 3-1과 같으며 크게 두 가지로 분류하여 평가한다. ①은 위키피디아로 학습된 Word2Vec 모델로 데이터의 동의어를 치환하여 증강된 데이터를 생성하는 과정이다. 이 과정에서 데이터는 두 배 증가하게 되어 학습 데이터로 사용된다. ② 데이터를 이용하여 LSTM/GRU 모델을 학습하는 과정이다. 여기서 데이터를 그대로 사용하거나 증강된 데이터를 사용하여 모델의 성능을 비교해 보았다. 학습이 완료된 모델과 선행연구의 기법을 학습 데이터에 적용해보고, 정확도 평가를 진행하여 가장 성능이 좋은 모델로 트위터 데이터에도 적용하여 비교해 보았다.

3.2.2 딥러닝 모델 구조

실험에는 LSTM, GRU 모델을 사용하여 각 성능을 비교하였다.

김윤(2014)은 “Convolution neural networks for sentence classification”에서 Convolutional layer를 이용하여 문서의 특성값을 추출하는 그림 3-2와 같은 모델을 제시했다.

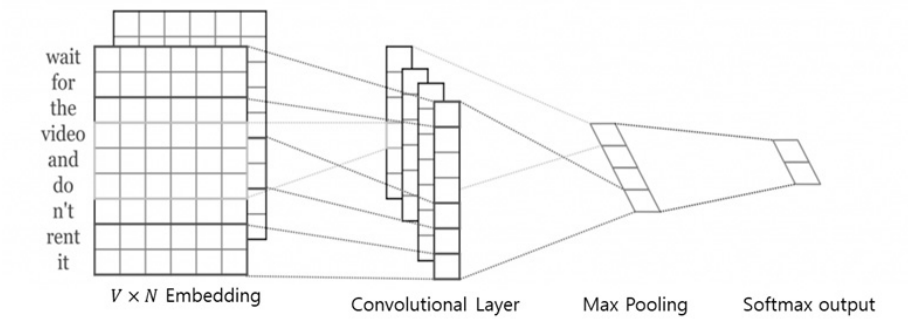


그림 3-2. CNN 기반 문장 분류 모델 모식도(Kim, 2014)

먼저, 벡터화된 문장을 각 단어에 대해 벡터 상의 공간에 임베딩한다. 이후, 여러 개의 필터를 이용해 슬라이딩하여 합성 곱(Convolution)을 통해 feature vector를 생성한다. 이후, Max pooling과 dropout을 이용하여 결과를 출력한다. 이 과정은 일반적인 CNN(Convolution Neural Network)의 일부와 같다. 위 과정을 통해 전체 데이터의 크기를 줄여 연산을 줄일 수 있으며, 과적합을 방지할 수 있게 된다.

Jiegzhan(2017)은 위의 Convolution-Max pooling 과정을 거쳐 생성된 각각의 feature vector들을 이용하여 RNN에 적용하여 텍스트를 분류하는 모델을 제안하였고, 본 연구에서도 같은 방식을 사용하였다.

Tensorflow에서 제공하는 시각화 툴인 Tensorboard로 나타낸 모델의 흐름도는 다음과 같다.

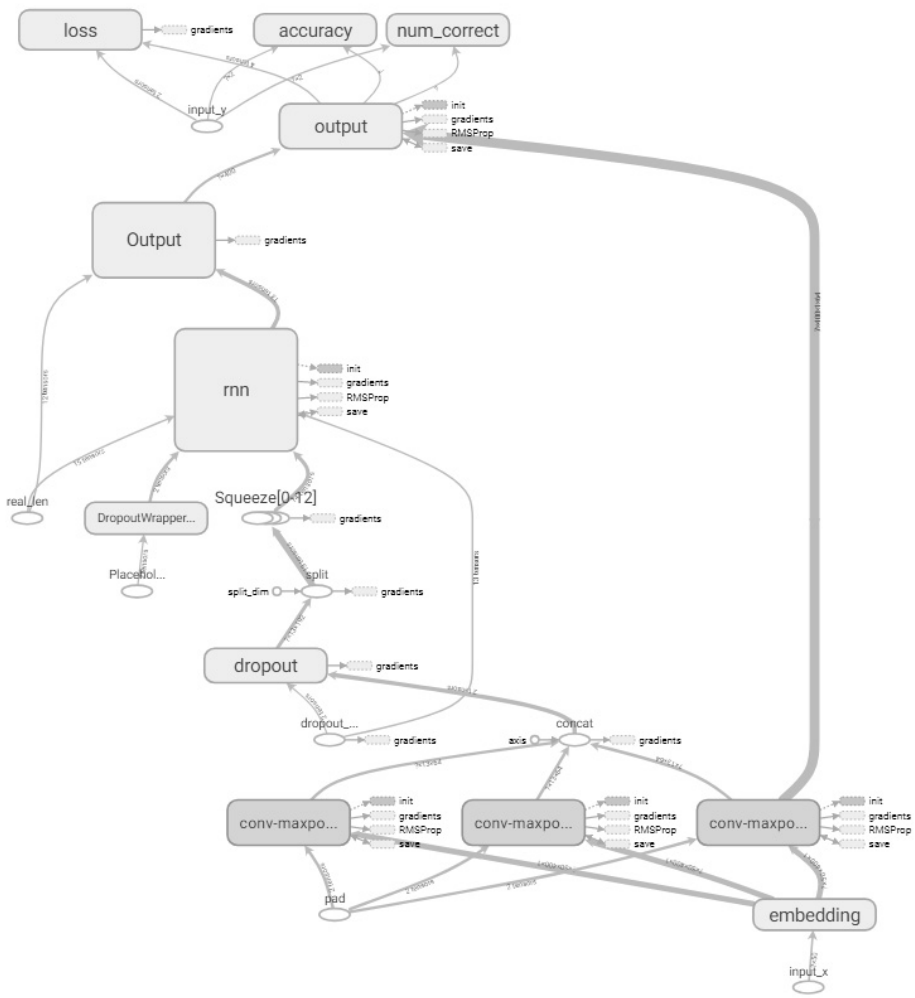


그림 3-3. Tensorboard로 나타낸 실험 모식도

위의 그림 3-3은 그림 3-4와 같은 흐름도와 표 2-4의 레이어로 설명할 수 있다.

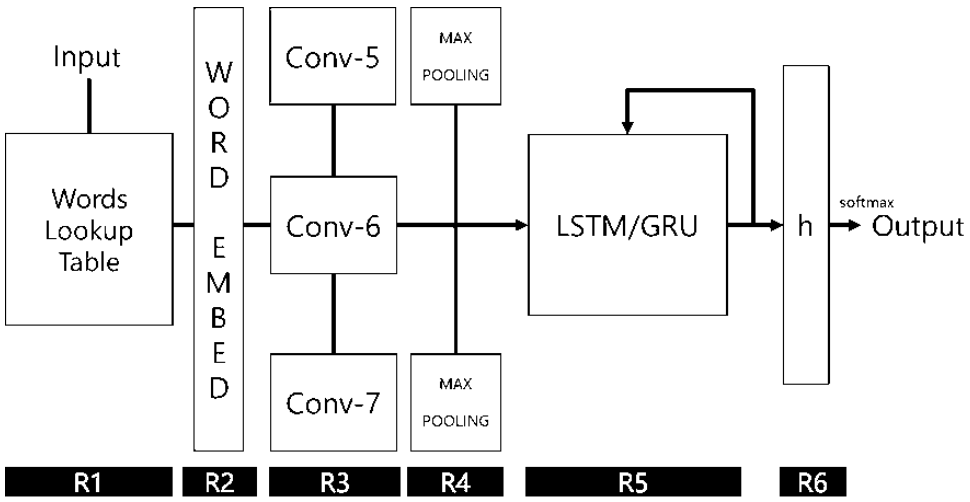


그림 3-4. 딥러닝 모델 레이어

표 3-4. 딥러닝 모델 레이어 설명

R1	R2	R3	R4	R5	R6
입력 값 (_×50)	워드 임베딩 (_×50×400)	3×Conv filter (_×50×1×64)	13×Max Pooling (_×13×1×64)	400× RNN units (_×400)	출력 값 (_×2)

※ _ : 입력된 문장 개수

본 연구에서 LSTM/GRU를 적용할 때에는 벡터화된 문장을 길이 50으로 자른 후, 400차원으로 워드 임베딩을 거친다. 입력값을 3개의 합성 곱 필터를 적용하여 Max pooling 하여 생성된 feature vector를 LSTM/GRU 레이어에 입력한다. LSTM/GRU는 최종적으로 손실(loss)과 예측한 라벨링 값을 출력하게 되며, 모델의 학습은 손실을 최대한 줄이는 방향으로 진행된다.

4. 실험 및 결과

4.1 데이터 처리

실험에는 Python 3.5, Tensorflow 1.6 버전을 사용하여 Windows 10 환경에서 사용하였다. 직접 라벨링 하여 모델 학습에 사용된 뉴스 문장 데이터는 5,931개이다.

한글 텍스트는 영어와 달리 명사가 조사와 결합에 따라 형태가 달라지기 때문에 Python의 KoNLPy(Korean Natural Language Processing in Python) 패키지를 이용하여 형태소 분석 후의 결과를 입력값으로 적용하였다. KoNLPy에는 품사 태깅을 위해 꼬꼬마, 코모란, 한나눔, 트위터 등의 분석기를 제공하고 있으며 각 분석기는 사용된 말뭉치와 품사 태깅 기준 및 말뭉치(corpus)에 조금씩 차이가 있다. 본 연구에서 각 분석기를 적용하여 실험해본 결과 코모란 분석기를 사용했을 때의 정확도가 가장 높았다. 또, 입력값에 대입하는 품사 태깅 요소에 따라 기계학습 모델의 성능이 달라진다는 선행연구(김민정, 2008)를 참조하여 여러 조합을 시도해 보았고, 모든 품사를 고려했을 때 가장 과적합 되지 않는 결과를 보여 실험에 적용하였다.

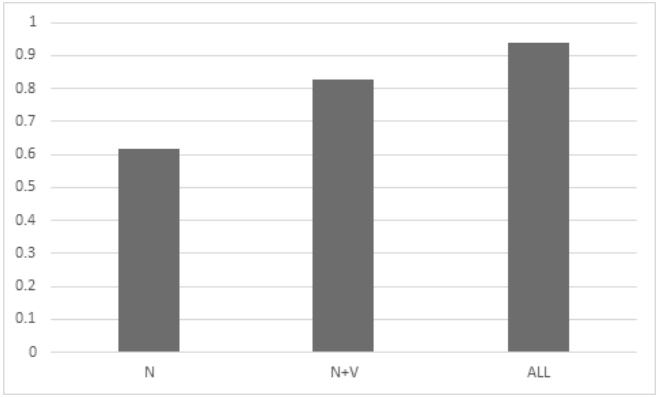


그림 4-1. 품사 태깅 요소별 모델의 성능

N은 명사(1), N+V은 명사와 동사(2), ALL은 모든 형태소(3)를 입력값으로 넣은 결과이며 F1-measure를 확인해본 결과 (3)의 경우가 가장 성능이 좋은 것을 확인할 수 있었다.

Word2Vec 모델은 텍스트 사이를 나타내는 임베딩 기법 중 탁월한 성능을 보이지만, 딥러닝을 진행할 때 성능에 크게 영향을 미치지 못한다는 조휘열(2015)의 연구가 있다. 따라서 본 연구에서 텍스트 임베딩을 진행할 때는 전체 문서에서 형태소 분석을 진행한 후 모든 형태소에 숫자를 부여한 다음 각 문장을 벡터화하는 lookup table 형식의 방법을 사용했다. 다음 표 4-1은 전체 데이터에 대해 형태소 분석 후, 생성한 사전(vocabulary)의 일부이다.

표 4-1. 형태소 사전 일부

형태소	Number
"<PAD/>"	0
“하/XSV"	1
"ㄴ/ETM"	2
"다/EC"	3
"에/JKB"	4
“이/JKS"	5
"가/JKS"	6
"을/JKO"	7
"었/EP"	8
"았/EP"	9
"고/EC"	10
"화재/NNG"	11
…	…

실험에 사용된 문장은 각각 길이가 달라 벡터화하기 위하여 길이를 같게 바꿀 필요가 있다. 실험 데이터의 문장은 형태소 분석 후 평균 50.31개의 형태소를 가지고 있어 각 문장을 길이 50인 벡터로 변환하였다. 여기서 "<PAD/>"는 문장의 길이를 맞춘 후, 생기는 여백에 부여한 값으로 일종의 패딩(padding)과 같다. 임베딩 후의 결과는 표 4-2와 같다.

표 4-2. 임베딩 결과 일부

Text	부산 동매산 중턱에서 50대 남성이 라이터를 켜고 휴대전화를 찾는 과정에서 불이 나 임야 1000㎡를 태우고 4시간 만에 진화됐다
	소방당국은 소방차 40대와 소방대원 134명을 현장으로 출동시켰고 오전 6시 53분쯤 불길을 완전히 잡았다
Embed	[390, 438, 2347, ..., 0, 0, 0]
	[1440, 939, 110, ..., 0, 0, 0]

각 문장은 위와 같은 형태로 벡터화되었고, 각 형태소는 400차원의 벡터에 랜덤하게 임베딩하여 의미 관계를 나타내었다. 형태소가 임베딩되는 차원이 클수록 더 다양한 의미 관계를 표현할 수 있지만, 계산량은 기하급수적으로 증가할 수 있다. 일반적으로 Word2Vec와 GloVe와 같은 대표적인 워드 임베딩 과정에서 300-500차원의 임베딩을 시행하기 때문에 본 연구에서도 300, 400, 500 차원으로 임베딩하여 적용해보았다.

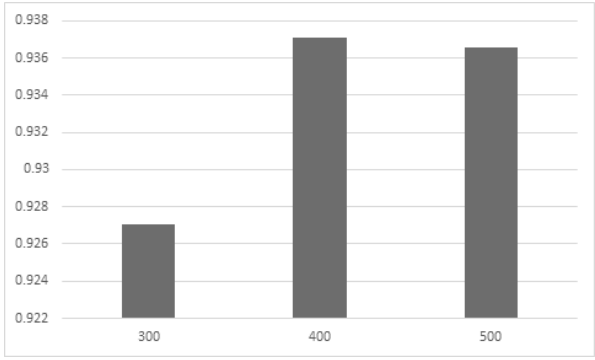


그림 4-2. 임베딩 차원별 모델의 성능

실험 결과, 400차원으로 임베딩 할 때 가장 성능이 좋음을 확인할 수 있었다. 이 결과로 보아 단순히 단어 사이의 정보를 많이 담는 것이 모델의 성능에 영향을 주지 않는 것을 확인할 수 있었으며 이후의 실험에도 같은 방법을 적용하였다.

데이터 증강을 위해서 Word2Vec 모델을 학습하였고, 데이터는 한국어 위키피디아에 제공하는 모든 문서를 사용하였다. Python의 gensim 패키지에서 제공하는 Word2Vec을 사용하였고, 직접 설정한 변수는 다음과 같다.

- 'min_count': 5 # 등장 횟수가 5 이하인 단어는 무시한다.
- 'size': 300 # 300차원짜리 벡터 스페이스에 임베딩한다.
- 'sg': 1 # 0이면 CBOW, 1이면 skip-gram을 사용한다.
- 'batch_words': 10000 # 사전을 구축할 때 한 번에 읽을 단어 수
- 'iter': 10 # 딥러닝에서 말하는 epoch, 반복 횟수

학습된 Word2Vec 모델을 이용하여 뉴스 문장 데이터를 동의어로 치환해보았다. 이때, 모든 형태소에 대해 치환이 진행하면 조사와 어미, 접미사까지 바뀌어버려 이해할 수 없는 문장이 생성되는 문제가 발생하였다. 따라서 명사에 한해서 동의어로 치환하였다.

동의어 치환은 most_similar 함수를 이용하여 표 4-3과 같이 동의어를 치환한 문장을 생성하였다. 1은 표적이 된 단어와 가장 유사한 단어로 바꾼 것이며 2는 두 번째로 유사한 단어로 바꾼 결과다.

표 4-3. 데이터 증강 결과 일부

	문장
원본	**경남** **밀양** **세종병원**에서 지난 26**일** **발생**한 **화재**로 39**명**이 **사망**했다 **오토바이**에 관한 한 **우리나라**는 **한참** **후진국**이다 **현장**에 있다 **경찰**에 **인계**된 A**씨**는 **라이터**를 켜 **바닥**에 떨어진 **휴대폰**을 찾다가 **낙엽**에 **불**이 붙었다고 **진술**한 **것**으로 알려졌다
1	**전북** **거창** **성모병원**에서 지난 26**개월** **일어나**한 **대화재**로 39**여명**이 **타계**했다 **모터사이클**에 관한 한 **한국**는 **며칠** **개발도상국**이다 **소방관**에 있다 **경찰관**에 **이관**된 A**김씨**는 **에디터**를 켜 **진흙**에 떨어진 **휴대전화**을 찾다가 **교복**에 **불길**이 붙었다고 **증언**한 **방법**으로 알려졌다..
2	**전남** **창녕** **이석행**에서 지난 26**일만** **빈발**한 **산사태**로 39**정**이 **서거**했다 **승합차**에 관한 한 **한반도**는 **잠시** **선진국**이다 **세월호**에 있다 **경찰서**에 **불하**된 A**정씨**는 **믹서**를 켜 **윗부분**에 떨어진 **핸드폰**을 찾다가 **관목**에 **비명**이 붙었다고 **자백**한 **만큼**으로 알려졌다

1에서는 일부 문법이 맞지 않는 것을 볼 수 있지만, 문장의 의미는 비슷하게 유지됨을 확인할 수 있다. 하지만 2의 결과를 보면 “산사태”와 같이 발생한 재난이 아예 달라지는 등, 재난의 종류나 문장의 의미가 바뀌어 버리는 경우가 다수 발생했다. 따라서 데이터 증강을 위한 동의어 치환 단계에서는 명사에 한해 가장 유사한 단어만 바꾸어 실험을 진행하였다. 데이터 증강 단계에서는 학습 데이터의 양이 원본 데이터의 두 배로 늘어나게 된다.

4.2 화재 재난 문장탐지 모델 평가

4.2.1 모델 평가

화재 재난 문장을 분류하기 위해 앞서 진행하여 증강된 데이터의 여부와 LSTM/GRU 알고리즘의 사용 방법에 따라 다음과 같이 다섯 가지로 나누어 실험을 진행해보았다.

① GRU
② LSTM
③ 데이터 증강+ GRU
④ 데이터 증강 + LSTM
⑤ 선행연구 기법

LSTM/GRU 모델에서 사용된 하이퍼 파라미터를 조정하며 실험해보았다. 실험 결과 convolution layer의 커널 크기를 세 개(3, 4, 5)로 놓고 batch size 20, hidden state의 수 400, max pooling size 4, learning rate 0.01, dropout ratio 0.5일 때의 성능이 가장 좋아 이후, 실험에 적용하였다. 경사 하강법을 구하기 위한 optimizer로는 Geoffrey Hinton(2012)의 RMSprop를 사용하였다.

5,931개의 데이터를 교차 검증(Cross-validation)하기 위하여 일반적으로 사용하는 Train/Test set 비율인 7:3으로 분류하였고, 과적합을 방지하기 위하여 Train set을 다시 7:3의 비율로 Train/Validation으로 나누어 동시에 평가를 진행하였다. 모델 ②, ③, ④, ⑤가 학습하는 과정 중의 정확도(Accuracy)와 손실(loss)을 Tensorboard를 이용하여 나타내 보았다. 각 모델에서 학습을 진행한 정확도와 손실 그래프는 다음과 같다. 주황색 선은 Training set, 빨강색 선은 Validation set에서의 결과를 나타낸다.

표 4-4. 모델 정확도 그래프

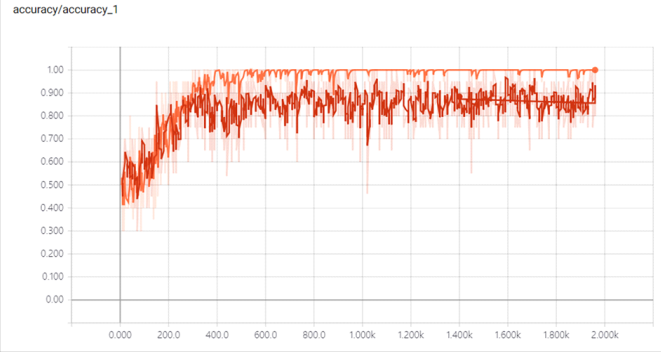
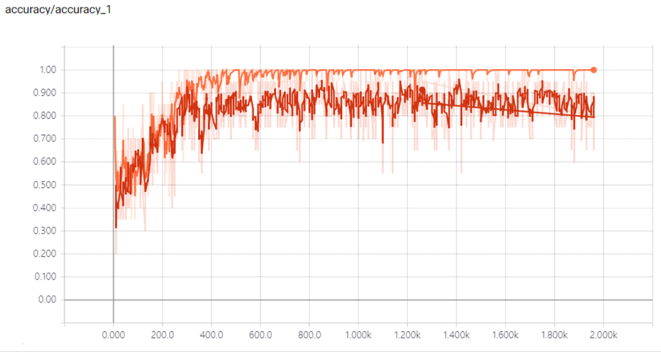
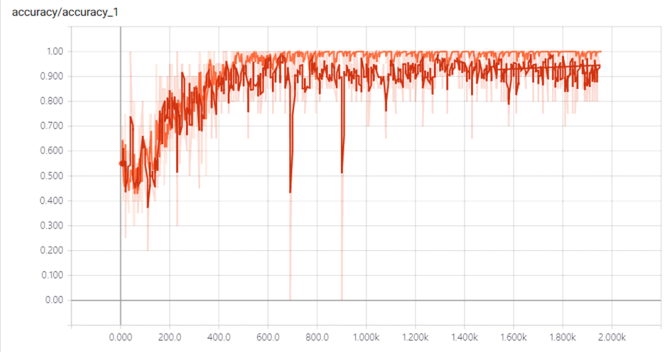
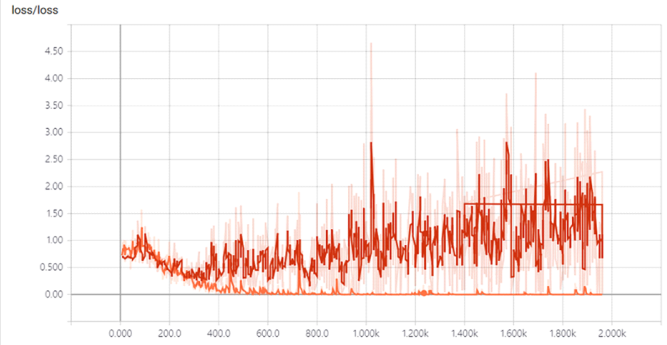
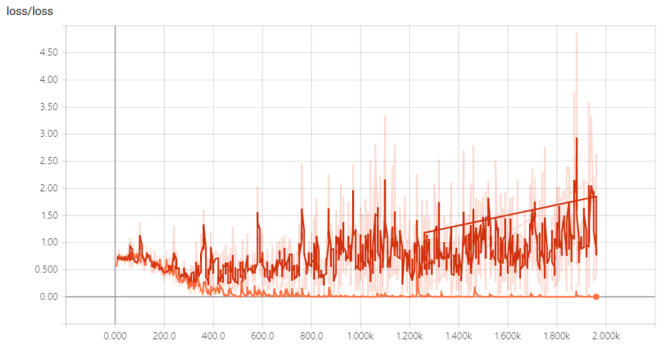
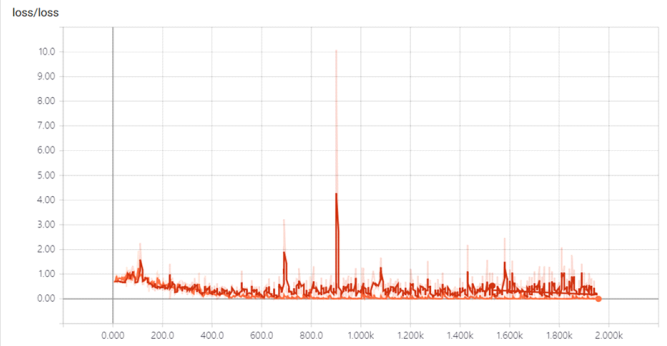
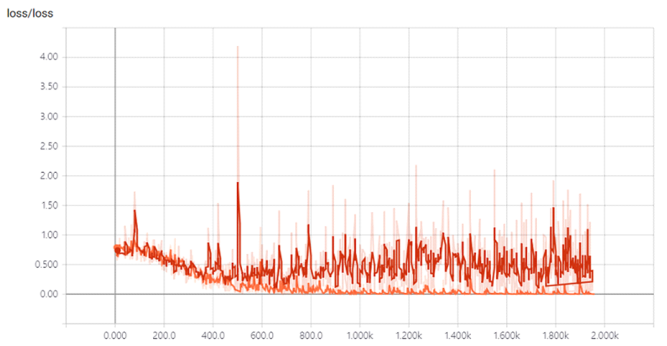
	GRU	LSTM
Original	① 	② 
Augment	③ 	④ 

표 4-5. 모델 손실 그래프

	GRU	LSTM
Original	① 	② 
Augment	③ 	④ 

①, ②의 정확도 그래프를 보면 Train set에서의 학습은 모두 이루어졌는데 Validation set에서의 정확도가 더 올라가지 못하고 있음을 확인할 수 있다. 데이터 증강 과정 후인 ③, ④ 모델의 Train set과 Validation set의 정확도가 더 차이가 나지 않고, 손실 값도 안정적으로 줄어드는 것을 확인할 수 있었고, 이는 데이터 증강 과정을 거친 후의 학습한 모델이 덜 과적합 되었고, 더 좋은 성능을 보였다고 해석할 수 있다. 데이터 증강을 거쳐 학습 데이터의 양이 2배 증가하였기 때문에 더 다양한 데이터를 통한 학습이 가능해져 과적합을 피했다고 추측할 수 있었다. 또한, 데이터 증강이 이루어지지 않았을 때 가장 최적의 모델은 Epoch = 1200 부근인 것으로 보아 그 시점에서 학습이 끝났음을 추측할 수 있다. 각 모델의 Test set에서의 정확도는 다음과 같았다. 정확도는 식 4-1과 같이 구하였다.

표 4-6. 각 모델의 Test set 정확도

모델	정확도
① GRU	0.9432
② LSTM	0.9321
③ 데이터 증강 + GRU	0.9525
④ 데이터 증강 + LSTM	0.9494

$$Accuracy = \frac{\text{올바르게 예측한 재난문장}}{\text{전체 재난문장}} = \frac{TP + TN}{TP + FP + FN + TN} \quad (4\text{-}1)$$

데이터 증강 과정을 거친 모델이 Test set에서 조금 높은 정확도를 보이는 것으로 보아 상술한 것과 같이 과적합을 피해 학습이 더 잘 이루어졌다는 해석을 내릴 수 있었다. LSTM과 GRU의 차이는 거의 없지만 GRU를 이용한 모델이 정확도가 더 높은 것을 확인할 수 있다. 데이터를 증강했을 때, LSTM의 정확도가 더 증가하는 것으로 보아 더 많은 입력값이 있을 때는 LSTM이 더 좋은 성능을 보일 것으로 보인다.

4.2.2 정확도 평가

표 4-7은 텍스트와 예측되어 나온 데이터 결과의 일부이다.

표 4-7. 탐지 모델 결과 일부

	Text	Actual	Predict
1	3일 오후 서울 마포구 홍대입구역 인근 서교동 사거리 신축공사장에서 화재가 발생해 연기가 피어오르고 있다	1	1
2	신차의 문제점도 분명히 있은 만큼 자동차 메이커의 소비자 중심에서의 역할과 정부의 전문가 집단 양성을 통한 중립적인 객관적 원인 확보도 중요한 준비 자세라 할 수 있다	0	0
3	이날 오전 3시께 중국집 배달원 유모(52)씨가 술에 취한 채 서울장여관을 찾아 "여자를 불러달라"며 성매매를 요구했다가 거절당하자 홧김에 불을 질러 투숙객 5명이 사망하고 5명이 부상당했다	1	0
4	그러나 소방법은 '화재 발생 시 출입문이 자동으로 닫히는 구조로 해야 한다'고 규정하고 이런 고정 장치 설치를 제한하고 있다	0	1

위의 표와 같이 라벨링 되는 결과를 확인할 수 있었고, 1과 2의 경우와 같이 기존의 분류를 제대로 예측하는 경우와 3, 4와 같이 잘못된 예측을 하는 경우로 나눌 수 있었다. 이는 표 4-8과 같이 나타낼 수 있다.

표 4-8. Confusion Matrix

		Actual Data	
		1	0
Predicted	1	True Positive(TP)	False Positive(FP)
	0	False Negative(FN)	True Negative(TN)

본 연구에서 제안한 모델과 선행연구의 성능을 평가하기 위해 식 4-2, 4-3, 4-4를 바탕으로 화재 재난 문장탐지 모델의 정확률(precision)과 재현율(recall)의 조화 평균인 F1-measure 값을 구하였다.

$$정확률(p) = \frac{TP}{TP+FP} \tag{4-2}$$

$$재현율(r) = \frac{TP}{TP+FN} \tag{4-3}$$

$$F_1 - measure = 2 \cdot \frac{p \cdot r}{p+r} \tag{4-4}$$

정확도 평가에 앞서 뉴스 데이터에서 학습에 사용되지 않는 2,004개의 데이터를 랜덤하게 추출한 후, 모델을 사용하여 재난 정보 문장을 찾아보았다. 그 결과는 아래의 표 4-9부터 4-13과 같다. 다섯 개의 모델 중 Augmented GRU 모델이 5,931개의 학습 데이터 중 높은 F1-measure인 0.653968을 나타냄을 확인할 수 있었고, 이는 기존 최선화(2015)의 패턴인식을 적용한 연구를 데이터에 적용한 결과인 0.477477보다 향상된 결과를 보인다.

표 4-9. ① GRU Confusion Matrix

		Actual		
		1	0	sum
Predicted	1	69	20	89
	0	89	1,825	1,914
	sum	158	1,845	2,003

표 4-10. ② LSTM Confusion Matrix

		Actual		
		1	0	sum
Predicted	1	120	169	289
	0	37	1,677	1,714
	sum	157	1,846	2,003

표 4-11. ③ 데이터 증강 + GRU Confusion Matrix

		Actual		
		1	0	sum
Predicted	1	103	54	157
	0	55	1,791	1,846
	sum	158	1,845	2,003

표 4-12. ④ 데이터 증강 + LSTM Confusion Matrix

		Actual		
		1	0	sum
Predicted	1	106	62	168
	0	52	1,783	1,835
	sum	158	1,845	2,003

표 4-13. ⑤ 선행연구 기법 Confusion Matrix

		Actual		
		1	0	sum
Predicted	1	106	180	286
	0	52	1,665	1,717
	sum	158	1,845	2,003

표 4-14의 결과와 같이 2,003개의 재난 정보 문장 실험 데이터에 대하여 모델의 학습 데이터에서의 분류 결과는 모든 모델이 패턴인식 기법보다 성능이 향상되었음을 보여준다. 또한, 딥러닝을 이용한 재난 문장탐지 기법은 사전 작성해야 하는 방대한 어휘 사전이 필요하지 않다. 따라서 패턴인식 기법보다 사용자의 간섭이 적다는 장점이 있고, 모델의 성능도 더 좋은 것으로 보여 재난 문장탐지 방법에 더 적합하다고 볼 수 있다.

표 4-14. 모델 평가

모델	Precision	Recall	Accuracy	F1-measure	AUC
① GRU	0.7753	0.4367	0.9456	0.5587	0.7129
② LSTM	0.4152	0.7643	0.8972	0.5381	0.8113
③ 데이터 증강 + GRU	0.6561	0.6519	0.9456	0.6540	0.8371
④ 데이터 증강 + LSTM	0.6310	0.6709	0.9431	0.6503	0.8186
⑤ 선행연구 기법	0.3706	0.6709	0.8842	0.4775	0.7866

분류 모델의 평가 지표인 ROC(Receiver Operating Characteristics) 커브의 면적인 AUC(The Area Under a ROC Curve)의 값은 1에 가까울수록 분류 모델의 민감도(sensitivity)와 특이도(specificity)가 모두 높으므로 우수한 모델이라고 할 수 있다. AUC의 값을 확인했을 때도 증강을 거친 GRU 모델이 가장 성능이 좋음을 알 수 있다. 이때, GRU 모델은 데이터 증강 과정을 거치지 않았을 때, 성능이 크게 떨어지는 것을 확인할 수 있어 증강 과정이 유의미하다고 볼 수 있다. 또 LSTM 모델의 경우, 학습에 필요한 파라미터의 수가 GRU 모델보다 매우 많아 데이터 증강 여부와 관계없이 모델의 성능이 크게 변화하지 않음을 보인다. 위의 결과로 학습 데이터가 충분히 크지 않을 때는 GRU 모델이 사용하기에 더 적합함을 알 수 있다. AUC와 F1-measure를 살펴보았을 때, 종합적인 성능이 모델 ③이 가장 우수하여 이후 트위터에 적용할 때 ③에서 학습된 모델을 이용하여 실험하였다.

4.3 트위터 데이터 적용 결과

트위터 데이터는 coordinate, user_id, text 등 30개의 속성값이 있으며 모든 속성값에 대하여 중복인 트위터를 제거한 후 2017년 12월 14일부터 12일 26일까지 트위터 364,477개를 수집하였다. 이 기간은 나무위키를 참조하여 최근 일어난 대화재 중 제천 스포츠센터 화재 발생 날짜인 2017년 12월 21일을 기준으로 약 10일간의 추이를 살펴보기 위하여 설정하였다. 데이터를 증강하고 GRU를 사용한 ③ 모델을 적용해 본 결과 총 3,081개의 트위터가 재난 정보 문장으로 탐지되었다. 다음 그림 4-11과 4-12는 날짜별 트위터 탐지개수와 날짜별 트위터 검출 비율을 나타내었다.

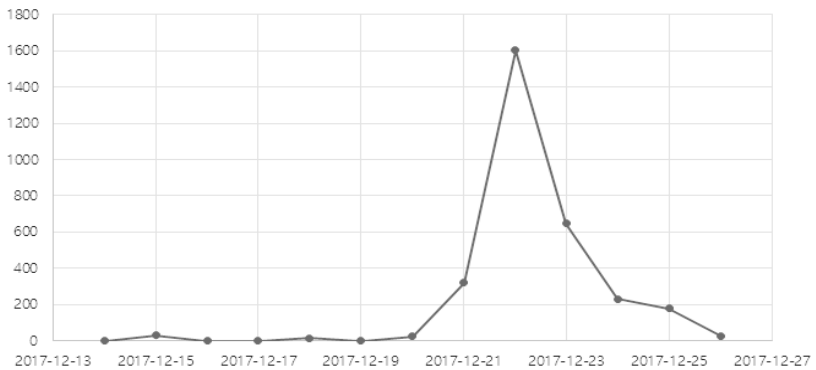


그림 4-3. 모델 ③ 날짜별 트위터 탐지개수

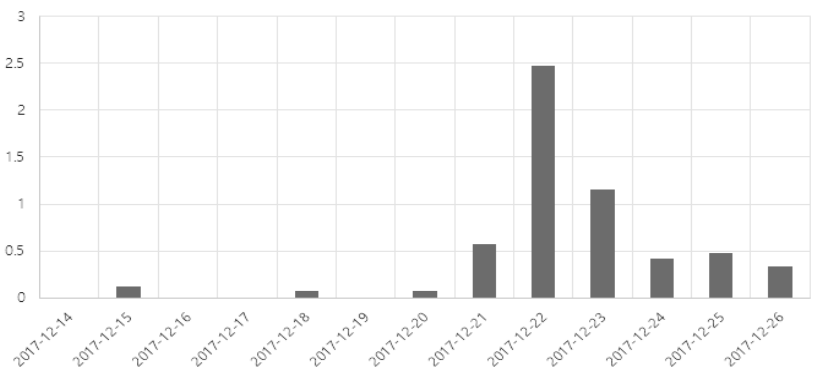


그림 4-4. 모델 ③ 날짜별 트위터 검출 비율(%)

그림 4-3을 보면 화재 재난이 발생한 21일 이후 탐지되는 개수가 늘어났다가 줄어드는 것을 확인할 수 있었다. 이는 재난 발생 후, 재난 정보 트위터에 대한 탐지가 잘 이루어졌다고 볼 수 있다. 그림 4-4에서도 화재가 발생한 21일 이후로 검출되는 비율이 늘어났다가 줄어드는 것을 확인할 수 있었다.

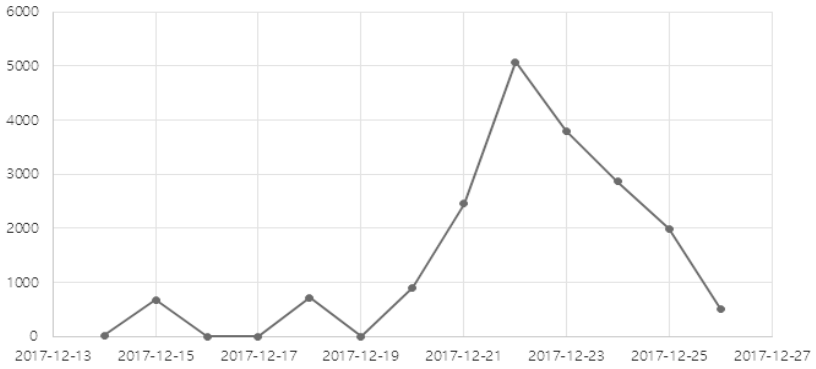


그림 4-5. 선행연구 ⑤ 날짜별 트위터 탐지개수

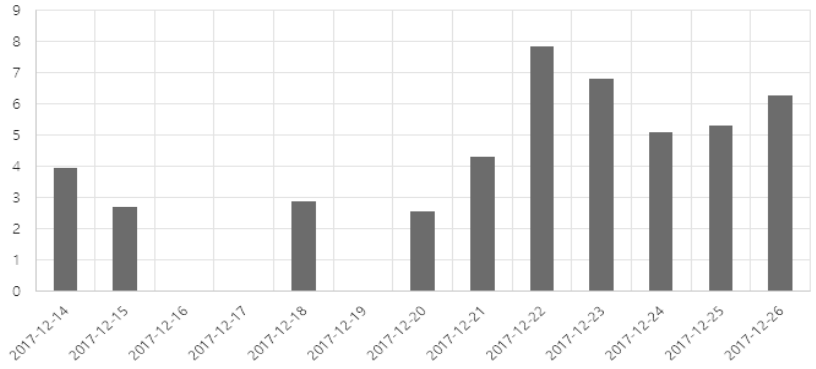


그림 4-6. 선행연구 ⑤ 날짜별 트위터 검출 비율(%)

그림 4-5, 4-6은 선행연구의 기법을 이용하여 트위터 데이터에서 검출 결과를 나타내었다. 그래프의 양상은 비슷하게 나오지만, 모델 ③의 결과가 항상 적게 나오는 것을 확인할 수 있었다.

정확도 평가를 위해 364,477개의 트위터를 모두 읽고 평가하기는 어렵다. 먼저, 트위터의 리트윗이라는 특성상 다른 사용자가 같은 텍스트를 언급하는 때도 있으므로 텍스트 내용이 중복된 결과를 제외하였다. 이후, 무작위로 추출한 3,000개의 트위터를 이용하여 평가하였다. 탐지된 문장 일부와 일부 표본에 대한 정확률은 표 4-15, 표 4-16과 같다. 표 4-15를 보면 잘못된 문장탐지의 경우, 딥러닝 모델은 화재와 전혀 관련 없는 문장을 탐지하는 것을 볼 수 있으며, 기존 패턴인식은 사용자가 오타나 잘못된 표현을 썼을 때를 구분하지 못하고 탐지하는 것을 볼 수 있다. 또한, 사용된 학습 데이터는 뉴스 데이터를 사용하였기 때문에 화재 발생에 대한 피해 상황을 구체적으로 전달하는 문장이 많아 “오전 2시 9분쯤 경찰이 도착했다.”, “이 사고로 버스에 타고 있던 시민 1명이 숨지고 15명이 다쳤다.”와 같은 문장들을 오 분류하는 경향을 보였다. 딥러닝 모델에서는 더 다양한 학습 데이터를 입력값으로 진행하면 해결할 수 있을 것으로 보인다.

표 4-15. 탐지된 재난 정보 트위터 일부

Text	Y	③	⑤
RT @ 불 번지는 속도 무섭다. 2층이 목욕탕 https://t.co/p5zTO8caTZ	1	1	1
후후 좋은 하루 보내시길~ㅎ	0	1	0
뭐야 이사람, 완전 내로남불아냐	0	0	1
장관 불러 놓고 야당 의원이 30분 가까이 호통을 칩니다. 법안 심사와 무관한 정치적 공격인데, 다른 상임위에서 통과된 법안의...	0	1	1

표 4-16. 모델 ③ 트위터 Confusion Matrix

		Actual		
		1	0	sum
Predicted ③	1	4	3	7
	0	6	2,987	2,993
	sum	10	2,990	3,000

표 4-17. 선행연구 ⑤ 트위터 Confusion Matrix

		Actual		
		1	0	sum
Predicted ⑤	1	10	69	79
	0	0	2,921	2,921
	sum	10	2,990	3,000

표 4-18. 트위터 데이터 모델 평가

모델	Precision	Recall	Accuracy	F1-measure	AUC
③ 데이터 증강 + GRU	0.5714	0.4	0.997	0.4709	0.6994
⑤ 선행연구 기법	0.1268	1	0.977	0.2247	0.9885

3,000개의 트위터 데이터의 평가 결과는 선행연구의 기법은 재현율이 100%가 나왔지만, F1-measure와 정확도는 제안한 모델이 더 높게 나오는 것을 확인할 수 있었다. 하지만 AUC의 값은 ③의 모델이 패턴을 이용한 기존 모델보다 매우 낮게 나옴을 확인할 수 있었다. 이러한 이유는 다음과 같이 추측해볼 수 있다.

첫 번째로 실제 데이터 안에 재난 정보 문장이 거의 없었기 때문에 탐지되는 문장 자체가 적어 생기는 오류이다. 본 연구에서 잘 학습된 모델이어도 학습 데이터에서 약 6% 정도의 오답률이 존재한다. 따라서 실험 데이터의 실제 재난 정보 문장 수가 적을수록 그 데이터에서의 성능은 떨어져 보인다. 본 연구에서 트위터 실험 데이터의 실제 재난 문장의 수는 10개뿐으로, 이러한 오류가 크게 나타난다. 두 번째로 모델이 학습 데이터에서 과적합이 일어나 그 외의 데이터에서의 성능이 저하되는 것으로 추측할 수 있다. 이는 학습 데이터의 양을 충분히 늘리는 것으로 해결 가능하며, 학습 데이터에서 더 다양한 형태의 재난 정보 문장의 수가 많을수록 더 좋은 성능을 보일 수 있다.

4.4 재난 문장탐지 기법 활용 방안

본 연구에서 제안하는 재난 문장탐지 기법은 빅데이터의 비정형 텍스트 데이터로부터 재난 정보를 가진 문장을 찾는 것을 가능하게 한다. 또한, 본 연구에서는 “화재” 재난의 문장을 위주로 탐지하였지만 다른 재난 정보 문장을 추가하여 학습에 활용한다면 여러 개의 재난 정보를 분류하여 탐지할 수 있는 시스템을 구축할 수 있다. 이는 재난 예방 분야에서 사용자를 일종의 센서로 사용하여 재난 예방 시스템을 구축할 수 있고, 재난 발생 후 정책을 수립할 때에도 다양한 의견을 수렴하거나 피해 상황 등을 파악할 수 있는 원시 데이터를 수집할 수 있다는 점에서 의의가 있다. 본 연구에서는 재난 문장탐지 기법을 이와 같은 재난 정보 서비스에 활용하는 방안을 살펴보았다.

현재, ETRI에서는 공공 인공지능 오픈 API를 포털을 통해 제공하고 있다. 인공지능 오픈 API는 언어 분석, 어휘 관계 분석, 질문 분석, 음성 인식과 같은 서비스를 제공하고 있으며 이 중 언어 분석 API의 개체명 인식(Named-entity Recognition, NER) 서비스를 사용하면 문장에서 개체명을 추출하여 비정형 데이터를 정형화된 형식으로 출력할 수 있게 된다.

그림 4-7. ETRI 공공 인공지능 오픈 API

개체명이란 인명, 지명, 기관명 등과 같은 특정 개체를 표현하는 단어를 말하며, ETRI에서는 15개의 대분류(인물, 학문 분야, 이론, 인공물, 기관, 지역, 문명, 날짜, 시간, 수량, 이벤트, 동물, 식물, 물질, 용어) 및 146개 세분류로 구성된 태그 셋을 바탕으로 개체명 인식 서비스를 제공하고 있다. 이 중 인공물(AF), 기관(OGG), 지역(LC), 날짜(DT), 시간(TI), 수량(QT) 등을 이용한다면 재난 정보에 대한 태그를 부착할 수 있다. 다음 표 4-17은 본 연구 실험 데이터에 적용한 일부이다.

표 4-17과 같이 텍스트로 이루어진 비정형 데이터에서 장소, 피해액, 시간, 날짜 등을 추출할 수 있었으며 개체명 인식 태그 부착 과정에서 일부 오류가 발생하는 것을 확인하였지만, API가 더 고도화된 후 적용한다면 현재 재난 안전 연구원에서 제공하는 재난 안전지도에 SNS를 분석한 실시간 재난 현황 등을 제공할 수 있게 된다.

표 4-19. ETRI OPEN API 적용 결과

Text	LC	TI
부산 동매산 중턱에서 50대 남성이 라이터를 켜고 휴대전화를 찾는 과정에서 불이 나 임야 1000㎡를 태우고 4시간 만에 진화됐다	['부산', '동매산']	['50대', '1000㎡', '4시간']
부산소방안전본부에 따르면 이날 오전 3시 6분쯤 부산 사하구 감천동 동매산 자락 중턱에서 불이 났다는 신고가 접수됐다	['부산', '사하구', '감천동', '동매산']	['부산소방안전본부', '오전 3시 6분']
소당당국은 소방차 40대와 소방대원 134명을 현장으로 출동시켰고 오전 6시 53분쯤 불길을 완전히 잡았다	NA	['소방차', '40대', '134명', '오전 6시 53분']
이번 산불로 임야 1000㎡를 태워 440만원 상당의 재산피해가 발생했지만 인명피해는 없었다	NA	['1000㎡', '440만원']

행정안전부의 국립재난안전연구원에서는 8가지 분야에 대해 안전지도를 제공하고 있다[3]. 그림 4-14와 4-15는 재난안전연구원에서 제공하는 생활 안전지도의 재난 안전 정보 제공 서비스화면과 실시간 정보 제공 서비스화면이다. PC와 모바일 환경에서 생활 안전 정보를 공개하여 제공하고 있으며 재난 안전 분야에서 화재 발생, 붕괴 발생, 산불 발생, 산사태 발생, 지진 발생 등의 재난 정보를 발생한 통계를 토대로 제공하고 있다. 또한, 실시간 정보로 통합대기지수, 미세먼지, 초미세먼지 등의 센서를 통해 입수할 수 있는 정보를 제공하고 있지만 본 연구의 재난 정보 문장탐지 모델과 인공지능 오픈 API 서비스를 결합하여 사용하면 실시간으로 SNS에서 발생하는 재난 정보 혹은 안전 신고 정보 등을 활용하는 것이 가능할 것이다.

그림 4-8. 생활안전지도 재난 안전 서비스화면

3) http://www.safemap.go.kr/main/smap.do?flag=2(방문일 : 2018.04.21)

그림 4-9. 생활안전지도 실시간 정보 서비스화면

5. 결론

인터넷에서 발생하는 데이터의 수가 점차 증가하고 있고, 빅데이터에서 좋은 성능을 나타내는 딥러닝 모델의 성능이 계속해서 발전하고 있다. 이로 인해 빅데이터에서 딥러닝을 이용하여 의사 결정에 의미 있는 정보를 줄 수 있는 연구가 계속해서 이루어지고 있다. 따라서 본 연구에서는 비정형 데이터에 딥러닝을 이용하여 재난 정보 문장을 탐지하는 모델을 제안하였고, 이는 텍스트 데이터에서 의미 있는 정보만을 판별할 수 있다는 점에서 의의가 있다.

기존의 사전 기반의 문장탐지 방식은 사용자가 많은 양의 사전을 직접 지속해서 관리 및 업데이트를 해주어야 하지만 딥러닝에서는 새로운 라벨링 데이터를 추가만 하면 모델이 스스로 학습하여 파라미터를 갱신한다. 따라서 본 연구에서는 딥러닝의 과적합을 막기 위해 데이터를 늘리는 과정에서 학습된 Word2Vec 모델을 통해 증강한 학습 데이터를 사용하고, 실제로 모델의 성능이 향상된 결과를 살펴볼 수 있었다. 이는 사용자의 개입을 최대한 줄여 기존의 방법 보다 효율적인 모델을 만들었다고 할 수 있다.

또한, LSTM 모델은 GRU 모델과 비교하여 파라미터의 수가 매우 많아 모델을 충분히 학습하는 데 필요한 데이터가 더 크다고 알려져 있다. 본 연구에서 각 모델의 AUC 면적을 비교한 결과 데이터의 증강이 이루어지지 않았을 때, GRU 모델의 성능이 선행연구보다도 떨어짐을 확인할 수 있었다. 따라서 GRU 모델이 과적합 되었음을 알 수 있었고, 데이터 증강 과정이 적절하게 이루어졌다고 판단된다. LSTM의 경우는 데이터 증강 여부와 관계없이 비슷한 성능을 보이므로 모델의 학습이 충분히 이루어지지 않았음을 알 수 있다.

본 연구에서 제안하는 문장탐지 모델을 ETRI의 인공지능 오픈 API와 결합하여 사용하면 비정형 데이터인 트위터 데이터에서 시공간적 정보와 재난의 피해 규모 또한 추출할 수 있게 될 것이다. 이 방법은 트위터를 비롯하여 모든 텍스트 데이터에 적용할 수 있으며, 화재 재난 외의 다른 재난 정보만이 아니라 원하는 정보를 가진 문장을 학습하게 되면 원하는 정보를 필터링하여 탐지하는 모델로도 사용할 수 있다. 이를 이용하여 현재 국립재난연구원에서 서비스하는 안전지도를 비롯한 위치 정보 서비스에 실시간 정보를 제공할 수 있을 것이다.

본 연구의 모델 정확도를 저하하는 요인은 다음과 같이 생각할 수 있다. 첫 번째, 연구자의 개입이 들어가는 라벨링 과정에서 혼자서는 더 많은 양의 학습 데이터를 생성할 수 없었다. 뉴스 데이터뿐만 아니라 트위터에 대해서도 라벨링을 진행하여 학습 데이터로 사용하였다면 트위터 데이터에서의 정확도도 더 향상되었을 것이다. 두 번째, 텍스트 데이터를 처리할 때 형태소 분석을 하게 되는데, 이 과정에서 맞지 않는 형태소로 태그를 하거나 하는 정확성이 떨어지는 문제가 있었다. 이는 전체적인 실험의 정확도를 떨어트리는 원인으로 형태소 분석기를 바꾸어가면서 실험을 해보았지만 조금씩 다른 오류가 계속해서 발생하였다. 이는 성능이 개선된 형태소 분석기를 이용하여 실험을 진행하면 해결할 수 있는 문제로 보인다.

최근에는 NIPS 2016(Conference on Neural Information Processing Systems 2016)에서 발표된 GAN(Generative Adversarial Network)과 같이 성능이 우수한 문장생성에 쓰일 수 있는 알고리즘이 존재한다. 추후 이러한 향상된 딥러닝 알고리즘을 사용하여 학습에 필요한 데이터를 재생성 하는데 적용하여, 재난 문장 탐지 기법의 정확도를 더욱 높일 수 있는 연구가 필요하다. 또한, 탐지된 문장에 담겨 있는 공간 및 시간 정보를 이용하여 시계열 분석을 진행하여 재난의 상관관계 및 요인분석에 대한 연구에 활용하고자 한다.

참 고 문 헌

김민정, 박재현, 김상범, 임해창, 이도길, 2008, 한국어 화행 분류를 위한 최적의 자질 인식 및 조합의 비교 연구. 정보과학회논문지: 소프트웨어 및 응용, 제35권, 제11호, pp.681-691.

김재훈, 최제영, 박충식, 2017, 빅데이터를 이용한 네트워크 이상탐지. 한국지능정보시스템학회 학술대회논문집, pp.158-159.

박성공, 2015, 재난 빅데이터 기반 전조감지 기술개발 및 소셜빅보드 확산전략 수립, 국립재난안전연구원, 국민안전처(보고서)

신동원, 2017, CNN-LSTM 복합모델을 이용한 대화의 발화 감정 분류, 고려대학교 대학원 석사 학위 논문.

유호선, 김현진, 오효정, 2018, 재난 사건별 이슈 생존 주기 유형 분석. 한국정보기술학회논문지, 제16권, 제3호, pp.126-135.

장경애, 박상현, 김우제, 2015, 인터넷 감정기호를 이용한 긍정/부정 말뭉치구축 및 감정분류 자동화, 한국정보과학회, 제42권, 제4호, pp.512-521.

조민희, 선충녕, 신성호, 엄정호, 홍승균, 송사광, 2015, 재난 이벤트 탐지를 위한 지식베이스 구축. 한국정보과학회 학술발표논문집, pp.733-735.

조휘열, 김진화, 윤상웅, 김경민, 장병탁, 2015, 컨볼루션 신경망 기반 대용량 텍스트 데이터 분류 기술. 한국정보과학회 학술발표논문집, pp.792-794.

조휘열, 2017, 딥러닝 기반 텍스트 질의응답을 위한 지식 추출 데이터 증강 기법, 서울대학교 대학원 석사 학위 논문.

조휘열, 김진화, 김경민, 장정호, 엄재홍, 장병탁, 2016, 순환 신경망 기반 대용량 텍스트 데이터 분류 기술. 한국정보과학회 학술발표논문집, pp.968-970.

최선화, 2016, 소셜미디어 위험도기반 재난이슈 탐지모델. 한국안전학회지 (구 산업안전학회지), 제31권, 제6호, pp.121-128.

하현수, 황병연, 2016, 트위터를 활용한 실시간 이벤트 탐지에서의 재난 키워드 필터링과 지명 검출 기법. 정보처리학회논문지. 소프트웨어 및 데이터 공학, 제5권, 제7호, pp.345-350.

Bengio, Y., Simard, P., & Frasconi, P., 1994, Learning long-term dependencies with gradient descent is difficult. IEEE transactions on neural networks, Vol.5, No.2, pp.157-166.

Bengio, Y., Ducharme, R., Vincent, P., & Jauvin, C., 2003, A neural probabilistic language model. Journal of machine learning research, Vol.3, pp.1137-1155.

Burel, G., Saif, H., Fernandez, M., & Alani, H., 2017, On Semantics and Deep Learning for Event Detection in Crisis Situations, In: Workshop on Semantic Deep Learning (SemDeep), at ESWC 2017, 29 May 2017, Portoroz, Slovenia.

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y., 2014, Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078.

Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. , 2014, Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint arXiv:1412.3555.

Earle, P. S., Bowden, D. C., & Guy, M. , 2012, Twitter earthquake detection: earthquake monitoring in a social world. Annals of Geophysics, Vol.54, No.6, pp.708-715.

Elman, J. L. , 1990, Finding structure in time. Cognitive science, Vol.14, No.2, pp.179-211.

Graves, A. ,2012, Supervised sequence labelling. In Supervised sequence labelling with recurrent neural networks , Springer, Berlin, Heidelberg. pp. 5-13.

Greff, K., Srivastava, R. K., Koutník, J., Steunebrink, B. R., & Schmidhuber, J. ,2017, LSTM: A search space odyssey. IEEE transactions on neural networks and learning systems, Vol.28, No.10, pp.2222-2232.

Hochreiter, S., & Schmidhuber, J., 1997, Long short-term memory, Neural computation, Vol.9, No.8, pp.1735-1780.

Karpathy, A., Johnson, J., & Fei-Fei, L. , 2015, Visualizing and understanding recurrent networks. arXiv preprint arXiv:1506.02078.

Kim, Y., 2014, Convolutional neural networks for sentence classification. arXiv preprint arXiv:1408.5882.

Lazard, A. J., Scheinfeld, E., Bernhardt, J. M., Wilcox, G. B., & Sura n, M., 2015, Detecting themes of public concern: a text mining analysis of the Centers for Disease Control and Prevention's Ebola live Twitter chat. American journal of infection control, Vol.43, No.10, pp.1109-1111.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J., 2013, Distributed representations of words and phrases and their co mpositionality. In Advances in neural information processing s ystems pp. 3111-3119.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estima tion of word representations in vector space. arXiv preprint ar Xiv:1301.3781.

Ruder, S., Ghaffari, P., & Breslin, J. G., 2016, Insight-1 at semeval-20 16 task 5: Deep learning for multilingual aspect-based sentime ntanalysis. arXiv preprint arXiv:1609.02748.

Wan, J., Wang, D., Hoi, S. C. H., Wu, P., Zhu, J., Zhang, Y., & Li, J. ,2014, Deep learning for content-based image retrieval: A com prehensive study. In Proceedings of the 22nd ACM internation al conference on Multimedia, pp. 157-166.

Yim, J., & Hwang, B. Y., 2015, TRED: Twitter based Realtime Event -location Detector. KIPS Transactions on Software and Data Engineering, Vol.4, No.8, pp.301-308.

Zhang, X., Zhao, J., & LeCun, Y., 2015, Character-level convolutional networks for text classification. In Advances in neural information processing systems, pp. 649-657.

Hinton, G., Srivastava, N., Swersky, K., 2012, Neural networks for machine learning lecture 6a overview of mini-batch gradient descent, https://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec6.pdf(접속일 : 2018년 3월 15일).

Jiegzhan, 2017, multi-class-text-classification-cnn-rnn, https://github.com/jiegzhan/multi-class-text-classification-cnn-rnn(접속일 : 2018년 2월 21일)

Neongen, 2018, 2nd place solution overview, https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge/discussion/52612(접속일 : 2018년 4월 20일)

Pu, C., Kitsuregawa, M., 2013, Big Data and disaster management: a report from the JST/NSF Joint Workshop. Georgia Institute of Technology, CERCS, https://grait-dm.gatech.edu/wp-content/uploads/2014/03/BigDataAndDisaster-v34.pdf(접속일 : 2018년 05월 2일)

Reinsel, D., Gantz, J., Rydning, J., 2017, Data Age 2025: The Evolution of Data to Life-Critical. Don't Focus on Big Data, http://www.seagate.com/www-content/our-story/trends/files/Seagate-WP-DataAge2025-March-2017.pdf(접속일 : 2018년 5월 2일)

Detecting Disaster Information Sentences from Unstructured Data Using Recurrent Neural Network - Case Study on Fire Accident -

Yim, Janghyuk

Department of Civil and Environmental Engineering

The Graduate School

Seoul National University

Abstract

Big data is composed of text which is unstructured data, and it is possible to analyze by using text mining to derive meaningful information for policy making and decision making. A recent studies using recurrent neural networks of text mining techniques show improved performance over CNN and other machine learning algorithms. In the deep learning algorithm, the efficiency and results of learning vary depending on the amount of high quality learning data. Therefore, in this study, by applying the technique data augmentation using the Word2Vec model, we tried to improve the accuracy of the disaster

sentence detection model. Furthermore, by comparing the results of the text analysis using LSTM and GRU is a kind of recurrent neural networks, we would like to improve the accuracy of the method of detecting a text containing information of fire on social media.

The disaster sentence detection model proposed in this research minimized user intervention and improved accuracy compared to previous research at the stage of data augmentation and model learning. In addition, the detected disaster sentence can be standardized using the object name recognition at a later date, which is meaningful in that it can extract information including disaster positions in unstructured data.

keywords : Deep Learning, Recurrent Neural Network, Data Augmentation, Word2Vec, Disaster Information, Social Media

Student Number **: 2016-24243**